



# Universidad de Cuenca

## Facultad de Ingeniería

### Carrera de Electrónica y Telecomunicaciones

---

## Diseño e implementación del sistema de instrumentación y comunicaciones de un exoesqueleto de extremidades inferiores

---

*Trabajo de titulación previo a la  
obtención del título de Ingeniero en  
Electrónica y Telecomunicaciones*

---

#### **Autores :**

Jonnathan Mauricio Blandín Segarra  
Anthony Javier Velasco Gordillo

C.I. 0105099949  
C.I. 0105972319

#### **Director :**

Ing. Luis Ismael Minchala Ávila, PhD.

C.I. 0301453486

#### **Co-Director :**

Ing. Andrés Marcelo Vázquez Rodas, PhD.

C.I. 0301496840

---

Cuenca - Ecuador  
15 de febrero de 2019



# Resumen

El riesgo de lesionar el aparato locomotor es común debido a la actividad física a la que están sometidas las extremidades inferiores, ya sea en actividades cotidianas, de recreación o de competición. En este sentido, para prolongar la independencia motora del individuo dentro de una vida normal, las terapias de rehabilitación son primordiales en la mantención e incluso en la mejora de las capacidades motoras. Hoy en día, existe un gran número de investigaciones interesadas en solucionar y reducir estas limitaciones traumatológicas, bajo el fundamento científico y desarrollo de tecnología, logrando así, la exploración de prototipos robóticos para la asistencia y rehabilitación llamados exoesqueletos.

El presente trabajo ofrece una alternativa a las terapias de rehabilitación basado en un sistema de instrumentación, control y comunicaciones de un exoesqueleto robótico para extremidades inferiores. El sistema de instrumentación se encarga de recolectar la información requerida a través de la red de comunicaciones formada por el protocolo CAN, para que el esquema de control PID tome decisiones en alusión a una trayectoria extraída de sujetos sin patologías. El diseño e implementación del dispositivo se aplica a la extremidad inferior izquierda y cuenta con tres grados de libertad correspondientes a las articulaciones de la cadera, rodilla y tobillo, las cuales se desplazan en el plano sagital. En esta primera versión se han ocupado cajas reductoras de tipo cicloidal y planetario para aumentar el torque en las articulaciones, siendo este primero de mayor relevancia por el aumento en el desempeño del control. Investigaciones futuras se enfocarán en la mejora de tasas de transmisión y tiempos de procesamiento.

**Palabras clave:** CAN. Cicloidal. Control. EPOS4. Exoesqueleto. Extremidades inferiores. Instrumentación. Planetario. PID. Protocolo. Reductores.





# Abstract

The risk of injury to the locomotor system is common due to the physical activity to which the lower extremities are subjected, whether in daily activities, recreation or competition. In this sense, to prolong the motor independence of the individual within a normal life, rehabilitation therapies are essentials in the maintenance and even in the improvement of motor skills. Nowadays, there are many researches interested in solving and reduce these traumatological limitations, under the scientific foundation and technology development, thus achieving the exploration of robotic prototypes for assistance and rehabilitation called exoskeletons.

This research work offers an alternative to rehabilitation therapies based on a system of instrumentation, control and communications of a robotic exoskeleton for lower extremities. The instrumentation system is responsible for collecting the required information through the communications network formed by the CAN protocol, so that the PID control scheme makes decisions in reference to a trajectory extracted from subjects without pathologies. The design and implementation of the device is applied to the left lower extremity and it has three degrees of freedom corresponding to the joints of the hip, knee and ankle, which move in the sagittal plane. In this first version, gearboxes of cycloidal and planetary type have been occupied to increase the torque in the joints, the first being of greater relevance due to the increase in the performance of the control. Future research will focus on improving transmission rates and processing times.

**Keywords:** CAN. Cicloidal. Control. EPOS4. Exoskeleton. Lower limbs. Instrumentation. Planetary. PID. Protocol. Reducers.





# Índice general

Resumen	III
Abstract	V
Índice general	VII
Índice de figuras	XI
Índice de tablas	XIII
Dedicatoria	XIX
Agradecimientos	XXI
Abreviaciones y acrónimos	XXIII
<b>1. Introducción</b>	<b>1</b>
1.1. Antecedentes . . . . .	1
1.2. Objetivos . . . . .	2
1.2.1. Objetivo general . . . . .	2
1.2.2. Objetivos específicos . . . . .	2
1.3. Estado del arte . . . . .	2
1.4. Contribuciones del trabajo de titulación . . . . .	6
<b>2. Fundamentos teóricos</b>	<b>7</b>
2.1. Robots usables . . . . .	7
2.1.1. Robots ortopédicos . . . . .	7
2.1.2. Robots protésicos . . . . .	8
2.2. Exoesqueletos . . . . .	9
2.2.1. Exoesqueletos para extremidades inferiores . . . . .	9
2.3. Biomecánica de las extremidades inferiores del ser humano . . . . .	10
2.4. Arquitectura de control . . . . .	12



2.4.1.	Regulación de corriente . . . . .	12
2.4.2.	Regulación de posición . . . . .	13
2.5.	Comunicación CAN . . . . .	15
2.5.1.	Descripción del protocolo CAN . . . . .	15
2.5.2.	Capa física . . . . .	16
2.5.3.	Capa de acceso al medio . . . . .	16
2.5.4.	CANopen . . . . .	17
2.6.	Módulo EPOS4 50/8 . . . . .	19
2.7.	EPOS Studio . . . . .	20
<b>3.</b>	<b>Diseño del exoesqueleto</b>	<b>21</b>
3.1.	Análisis cinemático del exoesqueleto . . . . .	21
3.1.1.	Modelamiento cinemático . . . . .	21
3.1.1.1.	Modelamiento cinemático directo . . . . .	22
3.1.1.2.	Modelo cinemático inverso . . . . .	23
3.2.	Análisis dinámico . . . . .	25
3.3.	Diseño mecánico . . . . .	28
3.3.1.	Reductores de velocidad . . . . .	28
3.3.1.1.	Reductor de la articulación de la cadera . . . . .	28
3.3.1.2.	Reductor de las articulaciones de rodilla y tobillo . . . . .	30
3.3.1.3.	Análisis de materiales . . . . .	32
3.4.	Diseño 3D . . . . .	33
<b>4.</b>	<b>Desarrollo del sistema de control, comunicación e instrumentación</b>	<b>35</b>
4.1.	Introducción . . . . .	35
4.2.	Señales de referencia . . . . .	37
4.3.	Conexiones y configuraciones entre motores y EPOS4 de cada articulación . . . . .	38
4.3.1.	Conexiones iniciales entre los motores y los EPOS4 . . . . .	39
4.3.2.	Configuración de los EPOS4 . . . . .	39
4.3.3.	Ajuste de constantes para el control PID . . . . .	43
4.3.4.	Topología de red implementada en el exoesqueleto para la comunicación . . . . .	44
4.4.	Determinación de los parámetros del sistema . . . . .	45
4.4.1.	Transformaciones de unidades . . . . .	46
4.4.2.	Interpolación parabólica . . . . .	48
4.5.	Programación para el control y comunicación del exoesqueleto de extremidad inferior . . . . .	49
4.5.1.	Ingreso de parámetros iniciales . . . . .	51
4.5.2.	Habilitar el EPOS4 . . . . .	53
4.5.3.	Posición inicial . . . . .	54
4.5.4.	Multiprocesos . . . . .	55



<b>5. Pruebas y resultados</b>	<b>57</b>
5.1. Evaluación de los algoritmos de control . . . . .	58
5.1.1. Control con constantes PID ingresadas manualmente . . . . .	59
5.1.2. Control con ajuste automático de las constantes PID . . . . .	62
5.1.3. Control con carga . . . . .	66
5.1.4. Evaluación del consumo energético y torque aplicado . . . . .	69
5.2. Evaluación del bus de comunicación . . . . .	71
5.2.1. Mensajes CAN . . . . .	71
5.2.2. Análisis de los retardos generados en el envío de mensajes CAN . . . . .	73
5.2.3. Cantidad de mensajes capturados . . . . .	75
<b>6. Conclusiones y Recomendaciones</b>	<b>77</b>
6.1. Conclusiones . . . . .	77
6.2. Limitaciones . . . . .	78
6.3. Recomendaciones . . . . .	79
6.4. Trabajos futuros . . . . .	80
<b>A. EPOS4 Module 50/8</b>	<b>83</b>
A.1. Pines utilizados del EPOS4 . . . . .	84
<b>B. Hojas de especificaciones técnicas de los motores EC 45 <i>flat</i> y EC 90 <i>flat</i> con <i>encoder</i> interno MILE</b>	<b>87</b>
<b>C. Conexión entre motores y los controladores EPOS4</b>	<b>93</b>
C.1. Distribución de pines y conexiones . . . . .	93
C.2. Diseño de placas de circuito impreso (PCB) . . . . .	96
<b>D. Instalación de la librería EPOS_Linux_Library</b>	<b>99</b>
<b>E. Planos de los diseños 3D</b>	<b>101</b>
<b>F. Planos de las cajas reductoras</b>	<b>107</b>
<b>Bibliografía</b>	<b>113</b>





# Índice de figuras

1.1. Prototipo Hardiman [1] . . . . .	3
1.2. Prototipo Hule [2] . . . . .	4
1.3. Prototipo Rex [3] . . . . .	5
1.4. Prototipo HAL [4] . . . . .	5
1.5. Prototipo Rewalk [5] . . . . .	6
2.1. Robot ortopédico para extremidades inferiores [6] . . . . .	8
2.2. Robot protésico para extremidades inferiores [7] . . . . .	8
2.3. Ciclo de caminata de un ser humano [8] . . . . .	10
2.4. División del ciclo de marcha [9] . . . . .	11
2.5. Arquitectura de control del EPOS4 [10] . . . . .	12
2.6. Regulador de corriente [10] . . . . .	13
2.7. Regulador de posición [10] . . . . .	14
2.8. Red básica para la comunicación CAN [11] . . . . .	15
2.9. Identificador de mensaje [11] . . . . .	18
2.10. Acceso al diccionario de objetos [11] . . . . .	18
2.11. Módulo EPOS4 50/8 [12] . . . . .	20
3.1. Inicio del modelado cinemático directo [9] . . . . .	22
3.2. Diagrama geométrico para el análisis cinemático inverso [9] . . . . .	24
3.3. Parámetros de los enlaces para el modelo dinámico [9] . . . . .	26
3.4. Elementos del reductor cicloidal [13] . . . . .	29
3.5. Diseño del disco cicloidal . . . . .	29
3.6. Elementos de un sistema de reducción planetario . . . . .	31
3.7. Resultados de la simulación en SOLIDWORKS . . . . .	33
4.1. Diagrama de bloques implementado . . . . .	36
4.2. Trayectorias de las tres articulaciones durante un ciclo de marcha . . . . .	38
4.3. Identificador y <i>bitrate</i> del EPOS4 colocado en la cadera . . . . .	40
4.4. Ingreso de parámetros iniciales del motor a utilizar . . . . .	41



4.5. Selección de sensores a utilizar . . . . .	41
4.6. Seleccionar el tipo de conmutación y el sensor . . . . .	42
4.7. Selección de tipo de control y sensor principal en el EPOS4 . . . . .	42
4.8. Selección de los límites de operación del EPOS4 . . . . .	43
4.9. Ajuste de constantes y prueba del control PID . . . . .	44
4.10. Topología de red utilizada en el exoesqueleto . . . . .	45
4.11. Relación de revoluciones en la cadera . . . . .	47
4.12. Relación de revoluciones en la rodilla y tobillo . . . . .	47
4.13. Perfiles de posición, velocidad y aceleración. (a) Gráficas con gran aceleración. (b) Gráficas con baja aceleración [14] . . . . .	49
4.14. Programación implementada en el exoesqueleto . . . . .	50
5.1. Exoesqueleto de extremidades inferiores . . . . .	58
5.2. Trayectorias de caminata con 10 s y con un ajuste manual de constantes de control . . . . .	60
5.3. Trayectorias de caminata con 7 s y con un ajuste manual de constantes de control . . . . .	61
5.4. Trayectorias de caminata con 4 s con un ajuste manual de constantes de control . . . . .	62
5.5. Trayectorias de caminata con 10 s y con un ajuste automático de las constantes de control . . . . .	63
5.6. Trayectorias de caminata con 7 s con un ajuste automático de las constantes de control . . . . .	64
5.7. Trayectorias de caminata con 4 s con un ajuste automático de las constantes de control . . . . .	65
5.8. Trayectorias de caminata con 10 s y con una carga total de 15 kg . . . . .	66
5.9. Trayectorias de caminata con 7 s y una carga total de 15 kg . . . . .	67
5.10. Trayectorias de caminata con 4 s y con una carga total de 15 kg . . . . .	68
5.11. Captura de mensaje SDO con <i>wireshark</i> . . . . .	71
5.12. Captura de mensaje SDO con <i>wireshark</i> . . . . .	74
5.13. Número de muestras con cada tiempo de caminata . . . . .	76
A.1. Distribución de pines en el EPOS4 . . . . .	85
C.1. Ubicación y distribución de pines de <i>Maxon Motor</i> EC45 con <i>encoder</i> . . . . .	93
C.2. Ubicación y distribución de pines de <i>Maxon Motor</i> EC90 con <i>encoder</i> . . . . .	94
C.3. Diseño de circuito impreso de placas del exoesqueleto . . . . .	98
D.1. Descarga de librería para EPOS4 . . . . .	99
D.2. Instalación de EPOS Library . . . . .	100





# Índice de tablas

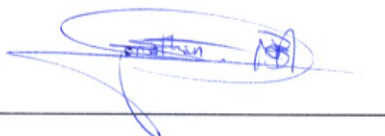
2.1. Características de la comunicación CAN [15]	16
3.1. Parámetros Denavit-Hartenberg para los enlaces del exoesqueleto	22
3.2. Rodamientos empleados en el reductor cicloidal	30
3.3. Descripción de los engranajes	32
3.4. Rodamientos empleados en el reductor planetario	33
4.1. Referencia de la cadera	37
4.2. Referencia de la rodilla	37
4.3. Referencia del tobillo	38
5.1. Consumo de corriente del exoesqueleto sin carga durante un ciclo marcha.	69
5.2. Consumo de corriente del exoesqueleto con carga durante un ciclo marcha.	69
5.3. Torque del exoesqueleto sin carga durante un ciclo marcha.	70
5.4. Torque del exoesqueleto con carga durante un ciclo marcha.	70
5.5. Mensaje enviado desde la Raspberry Pi 3 hacia la rodilla	72
5.6. Mensaje enviado desde la rodilla hacia la Raspberry Pi 3	72
5.7. Tiempo de transmisión sobre el bus CAN en comparación con el tiempo de procesamiento de la Raspberry Pi 3	73
A.1. Características eléctricas	83
A.2. Entradas y salidas	83
A.3. Interfaces	84
C.1. Conexión entre el motor y el controlador	94
C.2. Conexión entre el <i>encoder</i> y el controlador	95
C.3. Conexión entre el <i>encoder</i> externo y el controlador	95
C.4. Conexión del STO	95

## Cláusula de Propiedad Intelectual

---

Yo, Jonnathan Mauricio Blandín Segarra, autor del trabajo de titulación "Diseño e implementación del sistema de instrumentación y comunicaciones de un exoesqueleto de extremidades inferiores", certifico que todas las ideas, opiniones y contenidos expuestos en la presente investigación son de exclusiva responsabilidad de su autor.

Cuenca, 15 de febrero del 2019



Jonnathan Mauricio Blandín Segarra

C.I: 010509994-9

## Cláusula de Propiedad Intelectual

---

Yo, Anthony Javier Velasco Gordillo, autor del trabajo de titulación "Diseño e implementación del sistema de instrumentación y comunicaciones de un exoesqueleto de extremidades inferiores", certifico que todas las ideas, opiniones y contenidos expuestos en la presente investigación son de exclusiva responsabilidad de su autor.

Cuenca, 15 de febrero del 2019



Anthony Javier Velasco Gordillo

C.I: 010597231-9

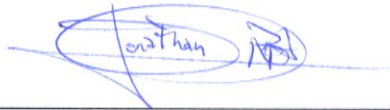
## Cláusula de licencia y autorización para publicación en el Repositorio Institucional

---

Yo, Jonnathan Mauricio Blandín Segarra en calidad de autor y titular de los derechos morales y patrimoniales del trabajo de titulación "Diseño e implementación del sistema de instrumentación y comunicaciones de un exoesqueleto de extremidades inferiores", de conformidad con el Art. 114 del CÓDIGO ORGÁNICO DE LA ECONOMÍA SOCIAL DE LOS CONOCIMIENTOS, CREATIVIDAD E INNOVACIÓN reconozco a favor de la Universidad de Cuenca una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra, con fines estrictamente académicos.

Asimismo, autorizo a la Universidad de Cuenca para que realice la publicación de este trabajo de titulación en el repositorio institucional, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

Cuenca, 15 de febrero del 2019



Jonnathan Mauricio Blandín Segarra

C.I: 010509994-9

## Cláusula de licencia y autorización para publicación en el Repositorio Institucional

---

Yo, Anthony Javier Velasco Gordillo, en calidad de autor y titular de los derechos morales y patrimoniales del trabajo de titulación "Diseño e implementación del sistema de instrumentación y comunicaciones de un exoesqueleto de extremidades inferiores", de conformidad con el Art. 114 del CÓDIGO ORGÁNICO DE LA ECONOMÍA SOCIAL DE LOS CONOCIMIENTOS, CREATIVIDAD E INNOVACIÓN reconozco a favor de la Universidad de Cuenca una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra, con fines estrictamente académicos.

Asimismo, autorizo a la Universidad de Cuenca para que realice la publicación de este trabajo de titulación en el repositorio institucional, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

Cuenca, 15 de febrero del 2019



Anthony Javier Velasco Gordillo

C.I: 010597231-9





# Dedicatoria

Dedico este trabajo de titulación a mi familia. Con un especial cariño a mi madre, Sarita, quien siempre me ha estado apoyando en todas las etapas de la vida. A Fernanda, quien me ha dado felicidad y soporte en momentos difíciles.

**Jonnathan Blandín**

Dedico este trabajo a mis padres Javier y Mariana, las personas más importantes en mi vida, que a pesar de todas las dificultades que se presentaron en el camino, ellos supieron alentarme con las palabras correctas y darme su apoyo incondicional. También a mis hermanos Xavier y Paola por la preocupación, aliento y confianza que supieron depositar en mí para seguir adelante.

**Anthony Velasco**







# Agradecimientos

La elaboración de este trabajo de titulación no ha sido únicamente esfuerzo de los autores. Varias personas han estado ligadas a este trabajo, las mismas que han contribuido de manera importante en la culminación del mismo. A todos ellos queremos expresarles nuestros más sinceros agradecimientos.

Primero a nuestras familias, por brindarnos el apoyo incondicional y sobre todo la paciencia a lo largo de la carrera y en la realización de este proyecto.

A nuestros directores, equipo de trabajo y amigos, los ingenieros Ismael Minchala, Andrés Vázquez, Fabián Astudillo, Esteban Mora, Jonnathan Peralta y Daniel Merchán, por ayudarnos con críticas constructivas, soluciones a los problemas que se presentaron y sobre todo por la motivación y apoyo entregado en la elaboración de este trabajo.

A los ingenieros, Lizandro Solano y Fabián Cabrera, por colaborarnos con un espacio tanto en el departamento de investigación como en el laboratorio de máquinas.

Finalmente, a cada uno de los profesores, quienes supieron transmitir sus amplios conocimientos y experiencias a través de las aulas universitarias, que de seguro serán herramientas muy útiles en el ámbito profesional.

**LOS AUTORES**





# Abreviaciones y Acrónimos

**AC** Alternating Current. 28

**BLDC** Brushless Direct Current. 19

**CAN** Controller Area Network. 2, 6, 15–18, 36, 39, 40, 45, 50, 52, 54, 57, 71–76, 78–80, 84, 85, 96, 98

**COB** Communication Object. 17

**DC** Direct Current. 19, 28

**DoF** Degrees of Freedom. 6, 22, 25, 28

**EC** Electronic Commutation. 28, 84, 87

**EMG** Electromyography. 2

**EPOS4** Easy Positioning System version 4. 12–14, 19, 20, 35–40, 42–45, 48, 50–56, 59, 64, 65, 69, 71–75, 83, 85, 94–96, 98, 100

**FDA** Food and Drug Administration. 4

**GUI** Graphical User Interface. 100

**HAL** Hybrid Asisted Limbed. 4

**HULC** Human Universal Load Carrier. 3

**IDE** Integrated Development Environment. 50, 51

**IEEE** Institute of Electrical and Electronics Engineers. 84

**ISO** International Standards Organization. 15, 16

**LLC** Logical Link Control. 16

**MAC** Media Access Control. 16

**Mbps** Mega Bit per Second. 15, 40, 45, 78

**MCUV** Movimiento Circular Uniforme Variado. 47



**OD** Object Dictionary. [17–19](#)

**PCB** Printed Circuit Board. [39](#), [80](#), [96](#)

**PDO** Process Data Object. [74](#), [76](#), [80](#)

**PID** Proportional Integral Derivative. [6](#), [13](#), [14](#), [36](#), [43](#), [44](#), [57–59](#), [61](#), [62](#), [64](#), [66–69](#), [78](#), [80](#)

**PLA** Polylactic Acid. [34](#)

**rpm** Revoluciones Por Minuto. [47](#), [48](#), [79](#)

**SCI** Spinal Cord Injury. [4](#)

**SDO** Service Data Object. [17](#), [18](#), [36](#), [45](#), [71](#), [72](#)

**SI** Sistema Internacional. [13](#), [14](#)

**STO** Safe Torque Off. [84](#), [95](#)

**USB** Universal Serial Bus. [40](#), [84](#)

# Capítulo 1

## Introducción

### 1.1. Antecedentes

La evolución tecnológica que se vive hoy en día junto con la inminente necesidad por ayudar a las personas con problemas traumatológicos, ha motivado la creación de nuevos equipos robóticos, que facilitan el trabajo médico y la recuperación de los pacientes. En este sentido, las patologías más comunes son las originadas en las extremidades inferiores. Esto principalmente debido a que toda la carga recae sobre éstas, y por ser además las responsables del desplazamiento, incluso en condiciones extremas [16]. Por esta razón, un gran número de estudios se enfocan en la rehabilitación del tronco inferior.

Los exoesqueletos, al ser básicamente estructuras para ser usadas sobre el cuerpo humano como apoyo para asistir a los movimientos y/o brindar capacidades ausentes del cuerpo, se ajustan de manera idónea en el campo de la salud. Comúnmente los exoesqueletos se adaptan al cuerpo mediante sistemas inteligentes de captura de señales biométricas y procesamiento de las mismas, y un sistema de control con el fin de que los actuadores correspondientes a cada articulación cumplan con su tarea.

De forma general, la operabilidad del exoesqueleto se fundamenta por la rama de instru-

mentación en conjunto con la rama de control, encargados de recolectar la información a ser enviada al procesador central para que, mediante estrategias de control, se tomen decisiones de manera acertada [17]. En adición, los campos de la física y la mecánica son importantes para modelar el movimiento y diseñar una estructura capaz de soportar a la persona.

En este contexto, el presente trabajo de titulación presenta el diseño e implementación de los sistemas de instrumentación, control y comunicaciones de un prototipo de un exoesqueleto robótico para extremidades inferiores. El sistema de control, considera como consigna señales de referencia extraídas de pacientes sin patologías. La carga útil del exoesqueleto se gestiona por un bus de comunicaciones establecido con una red de área controlada (CAN, por sus siglas en inglés), a la cual se conectan todos los dispositivos formando una arquitectura maestro-esclavo.

## 1.2. Objetivos

### 1.2.1. Objetivo general

Diseñar, desarrollar e implementar los sistemas de control e instrumentación de las extremidades inferiores de un exoesqueleto robótico.

### 1.2.2. Objetivos específicos

- Medir y estimar la posición y velocidad de las articulaciones de una extremidad inferior, además del sentido de rotación y torque de los motores.
- Diseñar un sistema de control de posición para los motores de las articulaciones.
- Transmitir la posición, velocidad y torque de las articulaciones como también las señales electromiográficas (EMG, por sus siglas en inglés) mediante la red basada en el protocolo CAN.
- Utilizar sistemas embebidos (microcontrolados) para la comunicación y control de las articulaciones.

## 1.3. Estado del arte

El estudio de los exoesqueletos robóticos empezó con proyectos que se inspiraron en el caparazón de algunos insectos y crustáceos. En un inicio, fueron desarrollados para incrementar la fuerza y resistencia de quien lo usara. En el año de 1965 la empresa General Electric comenzó a trabajar en un proyecto llamado Hardiman I [1]. Este prototipo fue construido y probado en

secciones separadas, comenzando con un solo brazo y una pierna, antes de ensamblar todo el exoesqueleto. Las pruebas de brazo único, como se puede ver en la Figura 1.1, fueron en gran parte exitosas, lo que le permitió levantar una carga de 750 libras. Las pruebas de piernas fueron más problemáticas, con dificultades para perfeccionar el mecanismo de retroalimentación y la movilidad. Estando totalmente ensamblado, el Hardiman I tendría una altura de 1.83 m, pesaría 1500 libras y usaría una combinación de servos hidromecánicos (manos) y electrohidráulicos (brazos y piernas) para el control y el movimiento. Sin embargo, después de años de desarrollo, el contrato del gobierno se agotó antes de que se pudiera completar un modelo completamente exitoso.



Figura 1.1: Prototipo Hardiman [1]

Con el avance del tiempo, se han mantenido los objetivos, pero las aplicaciones han aumentado significativamente. Por ejemplo, el ejército estadounidense desarrolló el proyecto [HULC](#) [2], un exoesqueleto hidráulico (Figura 1.2) destinado a ayudar a los soldados en combate a llevar una carga de hasta 200 libras, a una velocidad máxima de 10 millas por hora durante largos períodos de tiempo. Después de estar en desarrollo en Berkeley Robotics and Human Engineering Laboratory desde 2000, el sistema fue anunciado públicamente en el AUSA Winter Symposium el 26 de febrero de 2009 cuando se alcanzó un acuerdo de licencia exclusiva con Lockheed Martin. Aunque el exoesqueleto cuenta con energía autónoma y se puede usar, el proyecto fue un fracaso ya que obstaculizó ciertos movimientos y en realidad aumentó la tensión en los músculos, yendo directamente en contra de lo que se supone que debe hacer un exoesqueleto robótico.



Figura 1.2: Prototipo Hulc [2]

Entre las aplicaciones más recientes de los esqueletos se debe destacar a la rehabilitación de personas que han sufrido lesiones en sus extremidades. En este campo, REX Bionics [3] es la primera empresa en crear un exoesqueleto comercial, el REX, que puede mover a las personas con parálisis completa de la médula espinal. El REX (Figura 1.3) se puede soportar de forma independiente a sí mismo y al peso del usuario. Además, puede cambiar el peso de una pierna a la otra, caminar hacia adelante y hacia atrás, girar e incluso es capaz de subir escaleras. Cabe mencionar que tiene un precio en el mercado internacional de \$110 000 [18].

Existe otro prototipo llamado HAL para extremidades inferiores desarrollado por la empresa CYBERDINE Inc., el cual, es el exoesqueleto de rehabilitación médica móvil más ampliamente distribuido. Se trata de un robot de cadera y rodilla con un esquema de control de señal bioeléctrico que usa biosensores eléctricos adheridos a la piel para leer las señales generadas por la intención de movimiento (Figura 1.4). El precio en el mercado internacional es de \$96 000 [4].

Sin embargo, el primer prototipo en recibir la aprobación de la administración de alimentos y medicamentos (FDA, por sus siglas en inglés) para uso personal y de rehabilitación en los Estados Unidos es el ReWalk, un exoesqueleto robótico portátil que proporciona movimiento de cadera y rodilla para permitir que las personas con lesiones de médula espinal (SCI, por sus siglas en inglés) puedan pararse, caminar, girar, subir y bajar escaleras (Figura 1.5). En el mercado internacional tiene un precio de \$65 000 [5].





Figura 1.3: Prototipo Rex [3]



Figura 1.4: Prototipo HAL [4]



Figura 1.5: Prototipo Rewalk [5]

El desarrollo de la tecnología y el énfasis en este campo de investigación, darán origen a equipos más sofisticados haciendo que los prototipos en desarrollo sean más simples y de menor costo en comparación de los que se tiene actualmente.

## 1.4. Contribuciones del trabajo de titulación

El presente trabajo tiene como finalidad implementar un exoesqueleto de extremidades inferiores como una alternativa de rehabilitación, para que trabaje en conjunto con el campo de la salud y los profesionales encargados. Con este prototipo se piensa ayudar a personas que presenten diferentes patologías en su tronco inferior, enfocándose en mejorar su motricidad y fuerza.

Debido a que se cuenta con tres grados de libertad (**DoF**, por sus siglas en inglés) por pierna, se consigue reproducir los movimientos de la cadera, rodilla y tobillo, teniendo una rehabilitación total para las extremidades inferiores o en su defecto la dotación de movimiento para el paciente. El movimiento del exoesqueleto se lo logra con un control proporcional, integral y derivativo (**PID**) con características *anti-windup*, el cual sigue una trayectoria de referencia, la misma que ha sido capturada de personas sin patologías. Adicionalmente, en la transmisión de las tramas de datos, el sistema usa como capa física y de acceso al medio una red de comunicaciones **CAN**, siendo un protocolo comúnmente utilizado en entornos reducidos y con requerimientos de tiempo real. Mientras que a nivel aplicativo el protocolo usado es CANopen, diseñado originalmente para sistemas de control de máquinas orientadas al movimiento, como los sistemas de manejo.

## Capítulo 2

# Fundamentos teóricos

### 2.1. Robots usables

Los robots usables son un tipo específico de dispositivo que se utiliza para mejorar la movilidad y habilidades físicas de una persona. También son conocidos como robots biónicos o exoesqueletos [19]. Son robots orientados a ser utilizados por las personas. Pueden ser vistos como una tecnología que amplía, complementa, sustituye o mejora la función y capacidad de una extremidad humana. En otras palabras, son robots utilizados para complementar la función de una extremidad o reemplazarla por completo [20].

#### 2.1.1. Robots ortopédicos

Los robots ortopédicos son una estructura mecánica que se asocia a la anatomía de un miembro del ser humano. El objetivo es restaurar las funciones perdidas o débiles de alguna extremidad, complementando la capacidad de la extremidad humana y restableciendo la función de la incapacidad (Figura 2.1) [20].



Figura 2.1: Robot ortopédico para extremidades inferiores [6]

### 2.1.2. Robots protésicos

Los robots protésicos, por su parte, sustituyen las extremidades perdidas de un ser humano después de una amputación. Las prótesis adoptan la forma de extremidad robótica electromecánica y permiten reemplazar la función de la extremidad perdida de una forma más cercana a la función humana natural. Con la prótesis se ayuda a que la persona tenga una mayor autonomía sobre la extremidad perdida, además le permite tener una mejor integración social y laboral (Figura 2.2) [21].



Figura 2.2: Robot protésico para extremidades inferiores [7]

## 2.2. Exoesqueletos

Los exoesqueletos mecánicos son robots usables acoplados a las extremidades del cuerpo humano que están enfocados en el incremento de la fuerza, velocidad y rendimiento. Con los exoesqueletos se logra tener una transferencia directa de energía mecánica hacia las extremidades de una persona. Las principales aplicaciones se ubican en campos de la milicia, industria y medicina [22].

Un exoesqueleto puede además permitir que personas con problemas físicos tengan terapias de rehabilitación asistidas, guiando los movimientos de las trayectorias correctas para ayudar al paciente a recuperar los patrones de motricidad y dar fuerza a los movimientos que el individuo realice [23].

### 2.2.1. Exoesqueletos para extremidades inferiores

Las extremidades inferiores son las más vulnerables a sufrir lesiones y accidentes, lo que puede causar la pérdida de movimientos en las mismas. Por este motivo, la mayoría de exoesqueletos son diseñados para las extremidades inferiores.

Con los exoesqueletos para extremidades inferiores se puede asistir la marcha de personas parapléjicas o con problemas motrices. Además, se puede tener una mejora en rehabilitaciones de pacientes con problemas en sus piernas. Dentro de la rehabilitación, un exoesqueleto puede generar movimientos repetitivos y programados por un fisioterapeuta, permitiéndole medir el avance de mejora del paciente sin la necesidad de realizar de manera manual el trabajo pesado [24].

De acuerdo a [25] los exoesqueletos para extremidades inferiores se pueden dividir en tres grupos:

- Exoesqueletos para la rehabilitación de la marcha  
Son utilizados para personas que, por causa de alguna enfermedad cerebral, tienen atrofiados los músculos de sus extremidades inferiores. Las atrofas pueden ser generadas por golpes en la cabeza, parálisis cerebral o enfermedades infecciosas como el polio, lo que puede causar una pérdida total o parcial de la locomoción humana. Un exoesqueleto asiste el movimiento de un paciente, estimulando las extremidades inferiores y permitiendo a los fisioterapeutas que únicamente se concentren en el monitoreo y mejora del paciente en el tratamiento.
- Exoesqueleto para el aumento de fuerza  
Con este exoesqueleto se puede mejorar la fuerza que una persona sana tiene en sus extremidades inferiores. Permite que las personas realicen tareas que requieren mucho esfuerzo por si mismas, sin el soporte de otra persona de una manera sencilla. Es decir,

se amplifica la fuerza y resistencia de una sola persona. Este tipo de exoesqueletos son generalmente utilizados en la industria, en la milicia, en socorristas en caso de desastres, bomberos o en personas que requieren transportar cargas pesadas.

- Exoesqueletos para asistencia de la marcha

Este tipo de exoesqueleto ayuda a pacientes que tienen parálisis y tienen totalmente perdida la movilidad de sus extremidades inferiores. El exoesqueleto brinda el torque y la fuerza necesaria que el paciente necesita para moverse. La asistencia del exoesqueleto permite al paciente sentarse, pararse o caminar como una personas sana.

En este contexto, el presente proyecto propone el sistema de control y comunicación para un exoesqueleto de extremidades inferiores de la pierna izquierda, que permita realizar rehabilitación de la marcha.

## 2.3. Biomecánica de las extremidades inferiores del ser humano

La biomecánica es la disciplina que estudia el movimiento del cuerpo en diferentes circunstancias. Es decir, esta ciencia trata de analizar la actividad del ser humano y la respuesta que tiene el organismo ante esto [26].

La comprensión de la biomecánica en la caminata de un ser humano es importante para el diseño de exoesqueletos o robots ortopédicos para extremidades inferiores. En la Figura 2.3 se presenta el diagrama de un ciclo de caminata de un ser humano. Los porcentajes de tiempo presentados en la figura son una aproximación de la caminata, y pueden variar dependiendo del sujeto y sus condiciones específicas. El inicio del ciclo de la marcha se representa con el 0% y su final con 100% [8]

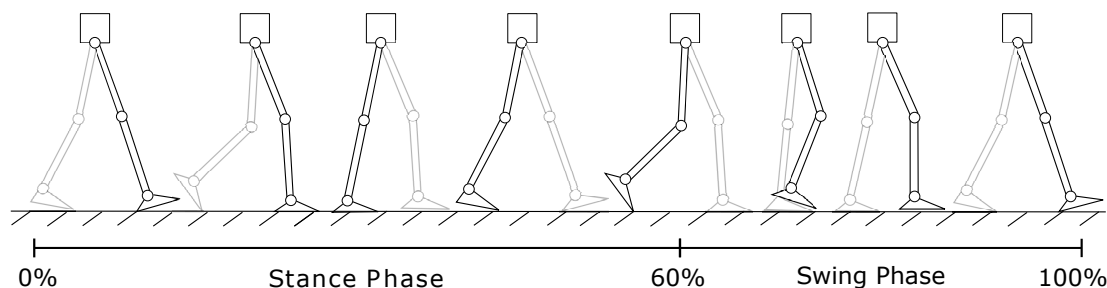


Figura 2.3: Ciclo de caminata de un ser humano [8]

La marcha es una actividad cíclica que empieza cuando el talón de un pie hace contacto con el suelo y termina con el siguiente contacto de talón del mismo pie. Un ciclo de marcha se divide en dos componentes principales: la fase de apoyo y la fase de balanceo [27]. Cuando una

pierna está apoyada al piso se dice que está en la fase de apoyo, y cuando la pierna se encuentra levantada y no toca el piso se dice que está en fase de balanceo. Las fases se van alternando de una pierna a la otra durante la marcha. La fase de apoyo ocupa el 60 % del ciclo de caminata y en el 40 % restante se cumple la fase de balanceo, como se puede ver en la Figura 2.3 [28].

La fase de apoyo está compuesta por 5 etapas:

- Contacto inicial: Es el instante en el que el talón toca el suelo.
- Puesta en carga: Contacto de la parte anterior del pie en el suelo.
- Apoyo medio: Momento en que el trocánter mayor se encuentra alineado verticalmente con el centro del pie, visto desde el plano sagital.
- Fase pre-oscilatoria: Momento en que los dedos se elevan del suelo.

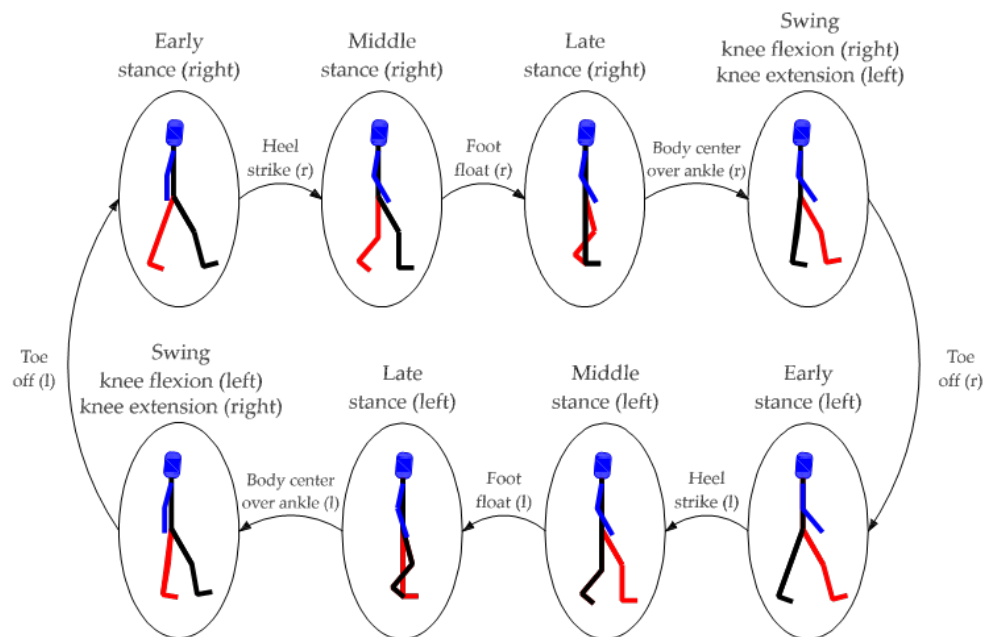


Figura 2.4: División del ciclo de marcha [9]

Es importante considerar que entre el contacto inicial y la puesta en carga, y el final de apoyo y la fase pre-oscilatoria se produce el apoyo doble, es decir, los dos pies tienen contacto con el suelo.

La fase de oscilación está formada por 3 etapas:

- Balanceo inicial: Se produce una rápida aceleración del extremo de la pierna, inmediatamente después de que los dedos dejan el suelo.
- Balanceo medio: La pierna en movimiento rebasa la pierna de apoyo como un péndulo.
- Balanceo final: La pierna desacelera al llegar al final del ciclo de marcha.

La biomecánica de un ciclo de marcha con cada una de sus etapas se muestra en la Figura 2.4. El movimiento de cada una de las articulaciones de la extremidad inferior genera variaciones angulares en las mismas, las cuales generan, a su vez, las trayectorias durante un ciclo de marcha. El valor de las variaciones angulares utilizadas para este proyecto se presentan en la sección 4.2. Estos son los valores acordes al movimiento de cada etapa de la Figura 2.4.

## 2.4. Arquitectura de control

Para comandar los motores de cada una de las articulaciones del exoesqueleto, se utiliza un controlador de los fabricantes *Maxon Motor*. El mismo se denomina *EPOS4* y estará encargado de mover los motores de acuerdo a la arquitectura de control implementada. La arquitectura del controlador *EPOS4* incorpora tres bucles de control, como se ilustra en la Figura 2.5:

- Regulación de corriente utilizado en todos los modos.
- Regulación de posición.
- Regulación de velocidad.

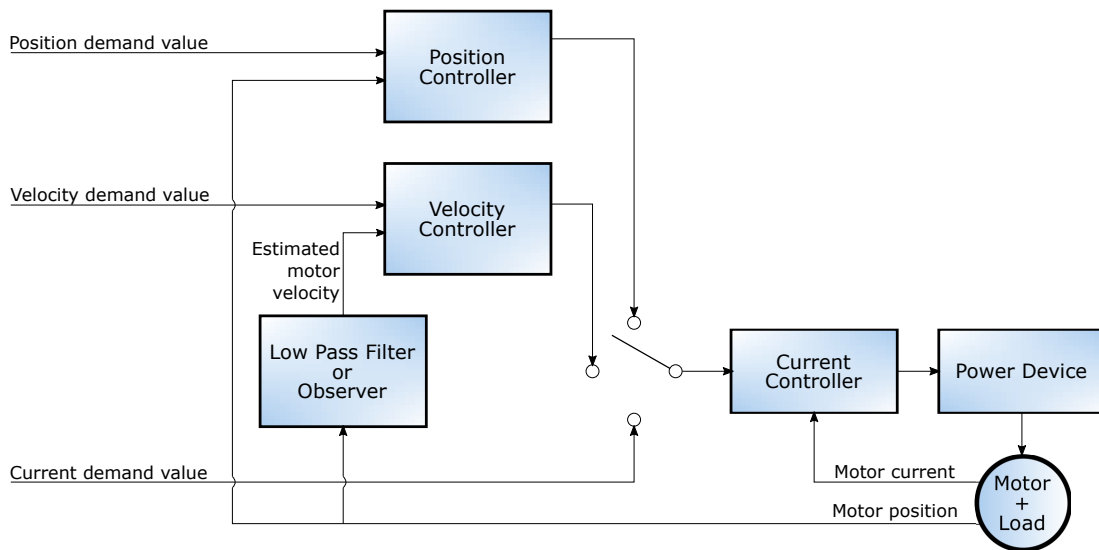


Figura 2.5: Arquitectura de control del *EPOS4* [10]

### 2.4.1. Regulación de corriente

Durante un movimiento dentro de un sistema de accionamiento, las fuerzas deben controlarse. Por lo tanto, como estructura de regulación principal, *EPOS4* ofrece un control basado en la corriente, tal como se muestra en la Figura 2.6.



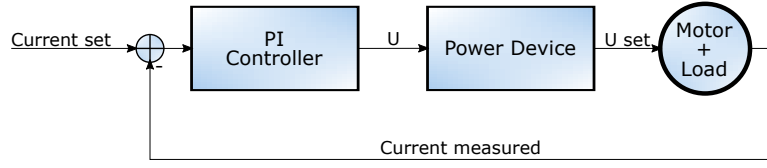


Figura 2.6: Regulador de corriente [10]

El controlador de corriente únicamente utiliza un control PI digital con un periodo de muestreo  $T_s = 0.04$  ms.

La unidades de las constantes de control son:

- $K_{P\_EPOS4} \rightarrow \left[ \frac{\text{mV}}{\text{A}} \right]$
- $K_{I\_EPOS4} \rightarrow \left[ \frac{\text{mV}}{\text{A} \cdot \text{ms}} \right]$

Las constantes están en unidades que puede entender el EPOS4. Si se desea transformar a las unidades del sistema internacional (SI, por su siglas en inglés) se utilizan las Ecuaciones 2.1 y 2.2. Una vez que se tienen las constante en unidades SI, se las puede utilizar para simulaciones numéricas o analíticas mediante la Ecuación 2.3.

$$K_{P\_SI} = 0.001 \cdot K_{P\_EPOS4} \quad (2.1)$$

$$K_{I\_SI} = K_{I\_EPOS4} \quad (2.2)$$

$$C_{corriente}(S) = K_{P\_SI} + \frac{K_{I\_SI}}{S} \quad (2.3)$$

Para evitar la degradación del rendimiento de control, cuando la entrada de control permanece en el valor límite durante períodos extendidos, se implementa un algoritmo anti-windup que impide que el componente integral del controlador PI tome valores mayores que los vinculados a la entrada de control.

El retardo de transporte total del bucle de regulación de corriente es siempre inferior a 0.06 ms.

### 2.4.2. Regulación de posición

EPOS4 posee un control de posicionamiento basado en un control de corriente subordinado. La Figura 2.7 muestra el diagrama de bloques de este esquema de control.

El regulador de posición utiliza un control PID digital con un periodo de muestreo  $T_s = 0.4$  ms. Además, tiene implementado un algoritmo de *anti-windup* que se utiliza para prevenir la acumulación del error integral dentro de control PID cuando los actuadores se saturan.

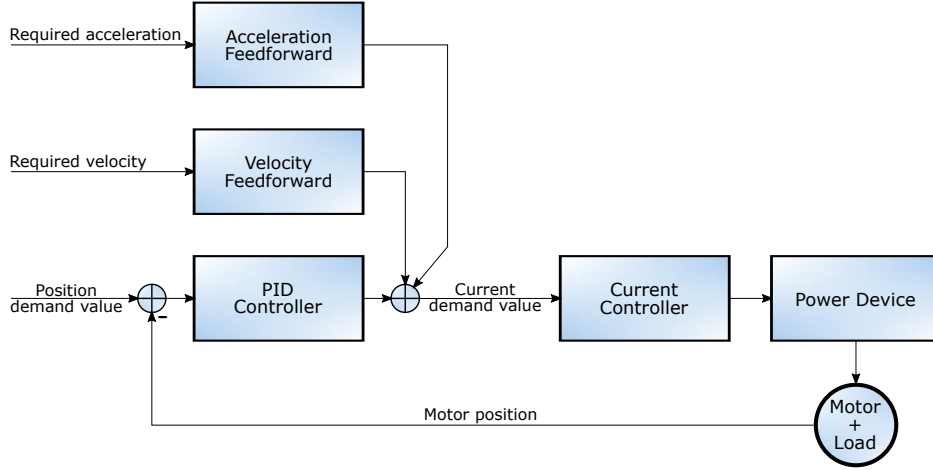


Figura 2.7: Regulador de posición [10]

Para mejorar el punto de referencia del sistema de movimiento, la regulación de posicionamiento se complementa con el control de pre-alimentación (*feedforward*). De este modo, el avance de la velocidad sirve para la compensación de la fricción proporcional a la velocidad, mientras que la anticipación de la aceleración considera la inercia conocida. Además, la parte diferencial de la señal del controlador PID se filtra con un pasa bajo antes de agregarse a los componentes proporcional e integral. El filtro pasa bajo se utiliza para evitar la influencia negativa en el rendimiento del control mediante la diferenciación de la posición del motor medida y ruidosa.

Las unidades de las constantes del control de posición son:

- $K_{PP\_EPOS4} \rightarrow \left[ \frac{\text{mA}}{\text{rad}} \right]$
- $K_{IP\_EPOS4} \rightarrow \left[ \frac{\text{mA}}{\text{rad} \cdot \text{s}} \right]$
- $K_{DP\_EPOS4} \rightarrow \left[ \frac{\text{mA} \cdot \text{s}}{\text{rad}} \right]$

De manera similar al controlador de corriente, el control PID se encuentra en unidades que entiende el EPOS4, y para transformar a unidades SI, se utiliza la Ecuación 2.4 . Para utilizar las constantes de control en simulaciones analíticas o numéricas se utiliza la Ecuación 2.5.

$$K_{xP\_SI} = 0.001 \cdot K_{xP\_EPOS4}, \quad \forall x = P, I, D \quad (2.4)$$

$$C_{posicion}(S) = K_{PP\_SI} + \frac{K_{IP\_SI}}{S} + \frac{K_{DP\_SI} \cdot S}{1 + \frac{K_{DP\_SI}}{10 \cdot K_{PP\_SI}} \cdot S} \quad (2.5)$$

El retardo de transporte total del bucle de regulación de posición siempre es inferior a 0,4 ms.

## 2.5. Comunicación CAN

La tecnología **CAN** es un protocolo de comunicaciones ampliamente utilizado en entornos y sistemas con requisitos de desempeño de tiempo real. Fue desarrollado por la compañía Robert Bosch GmbH en 1986, debido a la necesidad de conectar cada vez más dispositivos electrónicos en el interior de un automóvil [29]. El protocolo **CAN** utiliza un único bus de comunicaciones, el cual comparten todos los dispositivos electrónicos para evitar la conexión punto a punto entre cada uno de ellos (Figura 2.8).

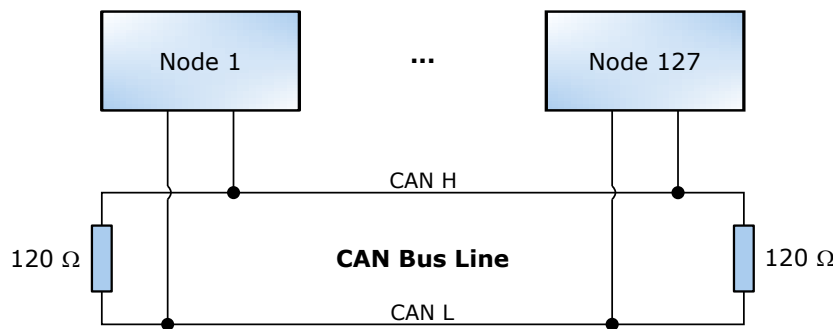


Figura 2.8: Red básica para la comunicación **CAN** [11]

La aplicación más común del bus **CAN** es la comunicación de los dispositivos electrónicos dentro de un automóvil como: computador a bordo, faros, sistema de frenos, control de temperatura, etc. Pero gracias a las ventajas de **CAN**, se ha empleado este protocolo en trenes, sistemas de aviación, control de ascensores y control de equipamiento médico, entre otros [30] [31].

### 2.5.1. Descripción del protocolo CAN

El protocolo **CAN** está especificado dentro del estándar **ISO 11898**, el cual tiene diversas normas específicas para distintos tipos de funcionamiento. Se tiene la norma **ISO 11898-2** para el protocolo **CAN** bus de alta velocidad, hasta 1 Mbps, y también la norma **ISO 11898-3** que corresponde al protocolo **CAN** bus de baja velocidad que también es tolerante a fallos [29]. Este proyecto utiliza la opción de bus **CAN** de alta velocidad. Los valores de tiempo y velocidad contemplados en **ISO 11898-2** se muestran en la Tabla 2.1.

De acuerdo a [29], el protocolo **CAN** presenta la siguientes ventajas:

- El protocolo **CAN** posee herramientas para la detección de errores en una transmisión y la retransmisión automática de tramas erróneas.

Tabla 2.1: Características de la comunicación CAN [15]

Velocidad	Tiempo de <i>bit</i> (us)	Longitud máxima (m)
1 Mbps	1	30
800 Kbps	1.25	50
500 Kbps	2	100
250 Kbps	4	250
125 Kbps	8	500
50 Kbps	20	1000
20 Kbps	50	2500
10 Kbps	100	5000

- Capacidad de discernir entre errores puntuales en la transmisión, o errores producidos por el fallo de un nodo, en cuyo caso, tiene la capacidad de desconectarlo para evitar que el error sature la red.
- Priorización de mensajes y garantía en los tiempos de latencia en la entrega de los mismos.
- Flexibilidad en la configuración de la red, tanto en cantidad de nodos, como en la disposición de los mismos.

### 2.5.2. Capa física

La capa física del protocolo CAN es la encargada de realizar la transferencia de bits entre los distintos dispositivos conectados en el bus. Es una línea de transmisión en topología de bus formado por dos hilos, con un retorno común que en ambos extremos termina con una resistencia para representar la impedancia característica de la línea. La resistencia a utilizar debe tener un valor de  $120\ \Omega$ . Los cables que forman el bus se denominan CAN\_H (CAN High) y CAN\_L (CAN Low), y con una diferencia de tensión entre los mismos de al menos 1.5 V [29], [32]. La red básica del bus CAN bus muestra en la Figura 2.8.

### 2.5.3. Capa de acceso al medio

La capa de acceso al medio de CAN también está estandarizada en la ISO 11898. Sus servicios se implementan en las subcapas de control de enlace lógico (LLC, por su siglas en inglés) y control de acceso al medio (MAC, por su siglas en inglés) de un controlador CAN [11].

- El LLC define la forma en que los datos son transferidos sobre el medio físico, proporcionando servicio a las capas superiores.
- La MAC es responsable de la encapsulación de datos, codificación de tramas, gestión de acceso al medio, detección de errores, señalización de errores, acuse de recibo y serialización.

El método de acceso al medio utilizado es el acceso múltiple por detección de portadora, con detección de colisiones y arbitraje de prioridad de mensaje (CSMA/CD + AMP). Cuando dos o mas nodos empiezan a transmitir mensajes al mismo tiempo el conflicto de acceso al bus de comunicaciones es resuelto mediante arbitraje usando un identificador. El identificador es utilizado para definir la prioridad de un mensaje, asegurando que el mensaje de mayor importancia sea transmitido primero [33].

#### 2.5.4. CANopen

El protocolo CAN únicamente define la capa física y de enlace de datos, así que es necesario definir cómo se asignan y utilizan los identificadores y datos de los mensajes CAN. Para ello se define el protocolo CANopen que implementa una capa de aplicación sobre la base de la comunicación CAN.

CANopen es un sistema de comunicación basado en CAN. Incluye protocolos de capa superior y especificaciones de perfil. CANopen se ha desarrollado como una red integrada estandarizada con capacidades de configuración altamente flexibles. Fue diseñado originalmente para sistemas de control de máquina orientados al movimiento, como la conducción automática de vehículos. Hoy se usa en diversos campos de aplicación, como equipos médicos, vehículos todoterreno, electrónica marítima, aplicaciones ferroviarias o automatización de edificios.

CANopen libera al desarrollador de tratar con detalles específicos del hardware de CAN, como la sincronización de bits y el filtrado de aceptación. Proporciona objetos de comunicación estandarizados (COB, por sus siglas en inglés) para procesos críticos de tiempo, configuración y datos de administración de red [34].

El concepto central de CANopen es el diccionario de objetos (OD, por sus siglas en inglés). OD es un grupo ordenado de objetos que describe completamente y de forma estandarizada la funcionalidad de cada dispositivo y permite su configuración mediante mensajes de objetos de datos de servicio (SDO, por sus siglas en inglés) a través del bus CAN [35].

Los servicios SDO se utilizan para acceder a la lista de objetos de un dispositivo conectado al bus. Es una conexión entre un cliente y un servidor SDO. El servidor es el propietario de los objetos que se desea acceder [36].

Es necesario definir el identificador del mensaje SDO. Éste está compuesto por 11 bits dividido en dos partes (Figura 2.9) [35]:

- 4 bits para el código de la función.
- 7 bits para el identificador de nodo (Node-ID).

Para el caso de SDO, los códigos de la función serán 1011 para transmitir y 1100 para recibir. El número máximo de nodos conectados a la red es de 127, el cual corresponde al

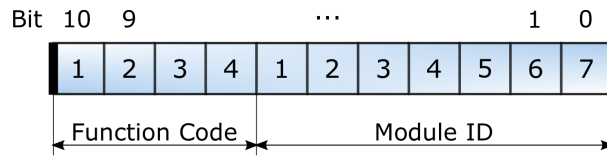


Figura 2.9: Identificador de mensaje [11]

campo de identificador del nodo. Al unir los dos campos se obtiene el identificador del OD. En formato hexadecimal puede tomar valores entre 0x601h - 0x67Fh para transmitir mensajes SDO y 0x581h - 0x5FFh para recibir mensajes SDO. Por este motivo, cada mensaje SDO involucra el intercambio de dos tramas CAN con diferentes identificadores [11].

Dentro de SDO, los mensajes siempre tienen una longitud de 8 Bytes, aunque no todos contengan información significativa (Figura 2.10). Para el caso de parámetros que contengan más de un Byte, se envía siempre el Byte menos significativo. Para formar las tramas de los mensajes SDO se necesita definir el término descarga, que significa escribir en el OD de un dispositivo y el término subida que es leer de él. Además, dentro de los dispositivos de comunicación que se utilizan en este proyecto, se maneja la transferencia expedita de mensajes, lo cual debe ser definido en la trama. Un cliente SDO transmite a su servidor utilizando el formato en la Figura 2.10 [35]:

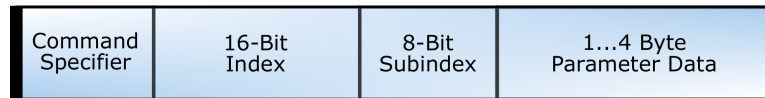


Figura 2.10: Acceso al diccionario de objetos [11]

- *Command Specifier* (1 Byte): Contiene información sobre si es subida o descarga de datos, petición o respuesta, transferencia expedita o en segmentos y el tamaño de los datos:
  - Especificador de comandos del cliente (3 bits): 001 para descargar y 010 para subida.
  - Ignorado (1 bit): se coloca 0.
  - Cantidad de Bytes (2 bits): cantidad de bytes que no contienen datos. Es válido si los siguientes dos bits son 1, caso contrario es 0. En subida se ignora.
  - Transferencia expedita (1 bit): Se coloca en 1 para indicar que es una transferencia expedita. En subida se ignora.
  - Tamaño de los datos (1 bit): Si se coloca 1 se debe especificar el tamaño de los datos. caso contrario se coloca 0. En subida se ignora.
- *Index* (2 Bytes): índice de la entrada del OD del servidor que el cliente desea acceder mediante el SDO actual. Este campo y el siguiente forman el multiplexor del dominio.
- *Subindex* (1 Byte): subíndice de la entrada del OD del servidor que el cliente desea acceder. Sólo tiene sentido si la entrada es de un tipo complejo. Si se trata de una entrada con tipo estático, debe ser 0.

- *Data* (4 Bytes): datos que se desean enviar al servidor. Si es subida se ignora.

El servidor al recibir el mensaje anterior, y si ha accedido con éxito al diccionario, contesta con un mensaje que tiene el siguiente formato:

- *Command Specifier* (1 Byte):
  - Especificador de comandos del servidor (3 bits): 011 para descarga y 010 para subida.
  - Ignorado (1 bit): se coloca 0.
  - Número de bytes (2 bits): número de bytes que no contienen datos. Es válido si los siguientes dos bits son 1, caso contrario es 0. En descarga se ignora.
  - Transferencia expedita (1 bit): Se coloca en 1 para indicar que es una transferencia expedita. En descarga se ignora.
  - Tamaño de los datos (1 bit): Si se coloca 1 se debe especificar el tamaño de los datos. caso contrario se coloca 0. En descarga se ignora.
- *Data* (4 Bytes): si es descarga se ignora ya que es una confirmación. Si es subida, contiene el valor leído del OD del servidor.

## 2.6. Módulo EPOS4 50/8

En el presente trabajo, para el control de cada una de las articulaciones, se utilizarán los módulos EPOS4 de *Maxon Motors*. El módulo EPOS4 50/8 (Figura 2.11) son unidades de control de posicionamiento inteligentes y digitales de pequeño tamaño. Su alta densidad de potencia permite un uso flexible para motores de corriente continua (DC, por sus siglas en inglés) y sin escobillas (BLDC, por sus siglas en inglés) de hasta aproximadamente 400 W. Tienen diversas opciones de retroalimentación, como sensores *Hall*, codificadores incrementales y sensores absolutos en una multitud de aplicaciones de accionamiento [12].

Los dispositivos son diseñados para ser comandados como un nodo esclavo ya sea en una red CANopen o EtherCAT. Estos también pueden ser operados mediante una comunicación serial vía USB o RS232 desde una estación que trabaje con sistemas operativos Windows o Linux. Las características técnicas más importantes se presentan en el Apéndice A.



Figura 2.11: Módulo EPOS4 50/8 [12]

## 2.7. EPOS Studio

*EPOS Studio* es un software disponible para Windows, que permite realizar la configuración inicial de todos los dispositivos EPOS de *Maxon Motor*. El software posee varios asistentes de ayuda que facilitan la configuración de los controladores como: la corriente, el número de pares de polos, y los sensores que va a utilizar el motor y el ajuste de las constantes de control de manera automática para que el EPOS4 funcione correctamente. Además, dentro de *EPOS Studio*, se dispone de herramientas que permiten el manejo de los motores de forma individual. De esta manera, asegurar el correcto funcionamiento de un motor y realizar ajustes adicionales si es necesario antes de colocar el motor en alguna estructura fija.

Es un *software* de libre descarga que se puede encontrar en la página de los fabricantes *Maxon Motor*. La instalación no solo contiene la interfaz de usuario de *EPOS Studio*, también incluye las versiones actuales de los *firmware* de todos los modelos de dispositivos EPOS y documentación importante sobre el manejo de cada uno [37].



## Capítulo 3

# Diseño del exoesqueleto

### 3.1. Análisis cinemático del exoesqueleto

#### 3.1.1. Modelamiento cinemático

Típicamente, el análisis cinemático es el punto de partida para el diseño de los sistemas robóticos. El modelado cinemático estudia el movimiento de los cuerpos, descartando las fuerzas que causan dicho movimiento. Principalmente entran en juego el modelado cinemático directo y el modelado cinemático inverso [38].

En el campo de la robótica, la cinemática directa, brinda la posición final de la estructura con base en los valores angulares de las articulaciones del sistema. Para este caso en específico, éstas corresponden a la cadera, rodilla y tobillo. Por otro lado, la cinemática inversa como se puede entender, actúa de forma contraria; es decir, al dar la posición final a la estructura, el modelado inverso se encargará de proporcionar los valores angulares necesarios de cada articulación para llegar a la posición objetivo [39].

### 3.1.1.1. Modelamiento cinemático directo

Como se mencionó en un principio, el prototipo cuenta con tres DoF. Se ha escogido este número con base a las articulaciones principales que tiene el miembro inferior correspondiente a una extremidad (cadera, rodilla y tobillo). La distribución de los DoF y los movimientos en el plano sagital se describen en la Figura 3.1a, obteniendo así el sistema coordenado que se ilustra en la Figura 3.1b, mediante el criterio de Denavit-Hartenberg [40].

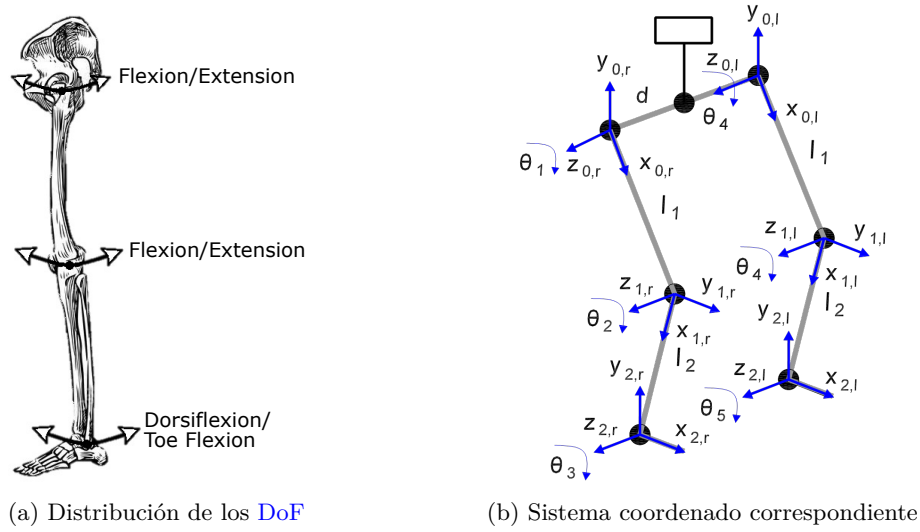


Figura 3.1: Inicio del modelado cinemático directo [9]

Con los sistemas coordenados se obtienen los parámetros de Denavit-Hartenberg, los mismos que se muestran en la Tabla 3.1.

Tabla 3.1: Parámetros Denavit-Hartenberg para los enlaces del exoesqueleto

Extremidad izq.	$a_i$	$\alpha_i$	$d_i$	$\theta_i$
1	$l_1$	0	0	$\theta_1$
2	$l_2$	0	0	$\theta_2$
3	$l_3$	0	0	$\theta_3$

El análisis se realizó considerando únicamente la extremidad izquierda en el plano  $xy$  (sagital), debido a la simetría existente. Con esto, la matriz de transformación que relaciona la posición del tercer enlace del miembro izquierdo está dada por la Ecuación 3.1.

$$T = \begin{bmatrix} C(\theta_1 + \theta_2 + \theta_3) & -S(\theta_1 + \theta_2 + \theta_3) & 0 & l_1 C\theta_1 + l_2 C(\theta_1 + \theta_2) + l_3 C(\theta_1 + \theta_2 + \theta_3) \\ S(\theta_1 + \theta_2 + \theta_3) & C(\theta_1 + \theta_2 + \theta_3) & 0 & l_1 S\theta_1 + l_2 S(\theta_1 + \theta_2) + l_3 S(\theta_1 + \theta_2 + \theta_3) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.1)$$

donde  $C(\theta_i)$  representa el coseno de la variable articular  $i$ , y  $S(\theta_i)$  representa el seno de la variable articular  $i$ .

La cuarta columna de la matriz  $T$  en la Ecuación 3.1, representa la posición del tercer enlace del miembro izquierdo. Por lo tanto, las ecuaciones cinemáticas directas del miembro izquierdo son:

$$\begin{aligned} x_d &= l_1 C\theta_1 + l_2 C(\theta_1 + \theta_2) + l_3 C(\theta_1 + \theta_2 + \theta_3) \\ y_d &= l_1 S\theta_1 + l_2 S(\theta_1 + \theta_2) + l_3 S(\theta_1 + \theta_2 + \theta_3) \\ z_d &= -d \end{aligned} \quad (3.2)$$

### 3.1.1.2. Modelo cinemático inverso

El modelamiento cinemático inverso utiliza el método de relaciones de triángulos. El análisis se realiza desde una vista sagital, debido a que en ese plano se obtienen los valores angulares principales para realizar una rutina de caminata. En la Figura 3.2 se puede observar la relación de triángulos implementada para este modelado con base en la definición de las siguientes variables:

- Posición de enlace final:  $[x, y]$
- Ángulo del tercer enlace:  $\delta$
- Longitudes de los enlaces:  $l_1, l_2, l_3$

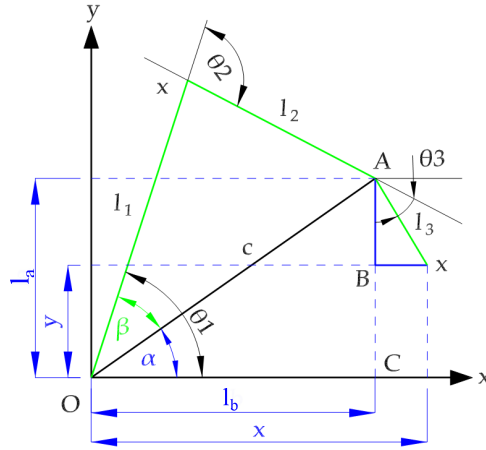


Figura 3.2: Diagrama geométrico para el análisis cinemático inverso [9]

Ahora, aplicando teoría de triángulos y proporciones, se tiene:

$$\begin{aligned} l_a &= y - l_3 \cos \delta \\ l_b &= x - l_3 \sin \delta \\ c &= \sqrt{l_a^2 + l_b^2} \end{aligned} \quad (3.3)$$

$$\alpha = \arctan \left( \frac{l_a}{l_b} \right)$$

$$\theta_1 = \alpha + \beta$$

$$\gamma = \arccos \left( \frac{l_1^2 + l_2^2 - c^2}{2l_1l_2} \right) \quad (3.4)$$

$$\theta_2 = \gamma - \pi$$

$$\theta_3 = \delta - \theta_1 - \theta_2$$

Finalmente, sustituyendo las variables  $\alpha$  y  $\beta$  en  $\theta_{1,2,3}$ , se tiene:

$$\begin{aligned} \theta_1 &= \arctan \left( \frac{y - l_3 \cos \delta}{x - l_3 \sin \delta} \right) + \arccos \left( \frac{l_1^2 + l_2^2 - (y - l_3 \cos \delta)^2 - (x - l_3 \sin \delta)^2}{2l_1 \sqrt{(y - l_3 \cos \delta)^2 - (x - l_3 \sin \delta)^2}} \right) \\ \theta_2 &= \arccos \left( \frac{l_1^2 + l_2^2 - (y - l_3 \cos \delta)^2 - (x - l_3 \sin \delta)^2}{2l_1l_2} \right) \\ \theta_3 &= \delta - \theta_1 - \theta_2 \end{aligned} \quad (3.5)$$

### 3.2. Análisis dinámico

Existen dos metodologías bien conocidas para obtener el modelo dinámico de cualquier mecanismo. El método de Euler-Lagrange, que es un método de energía, y el método de Newton-Euler, que se basa en el equilibrio de fuerzas y pares. En lo sucesivo, el método Euler-Lagrange se usa para obtener las ecuaciones dinámicas del exoesqueleto, cuyas ecuaciones son:

$$\begin{aligned}\tau &= \frac{d}{dt} \left( \frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} \\ L &= K - P\end{aligned}\tag{3.6}$$

donde  $i$  representa el DoF,  $q_i$  representa las coordenadas generalizadas (coordenadas articulares  $\theta_i$ ),  $\tau$  representa el vector de fuerza y torque aplicado a la articulación  $i$ ,  $L$  es la función Lagrangiana,  $K$  es la energía cinética del sistema y  $P$  es la energía potencial del sistema.

Existen dos tipos de modelos dinámicos:

- Modelo dinámico directo: expresa la evolución transitoria de las coordenadas articulares del robot, de acuerdo con las fuerzas y pares involucrados en el sistema.
- Modelo dinámico inverso: expresa las fuerzas y pares en cada unión del sistema, de acuerdo con la evolución transitoria de las coordenadas articulares y sus derivadas.

El modelo dinámico directo se obtiene a continuación, comenzando con el cálculo de la energía cinética,

$$K_i = \frac{1}{2} m_i \dot{\mathbf{s}}_i^T \dot{\mathbf{s}}_i + \frac{1}{2} \boldsymbol{\omega}_i^T \mathbf{I}_i \boldsymbol{\omega}_i\tag{3.7}$$

donde  $m_i$  es la masa del enlace  $i$ ,  $\dot{\mathbf{s}}_i$  es la velocidad traslacional del centro de gravedad  $G$  del enlace  $i$ ,  $\mathbf{I}_i$  es la matriz de inercia del enlace, y  $\boldsymbol{\omega}_i$  es la velocidad angular.

La energía potencial de un enlace  $i$  es definido por:

$$P_i = m_i g h_{ci}\tag{3.8}$$

donde  $m_i$  es la masa del enlace,  $g$  es la aceleración de la gravedad,  $h_{ci}$  es la distancia entre el origen del enlace y el centro de gravedad del enlace  $i$  paralelo al vector de gravedad.

Por lo tanto, se obtiene la siguiente función Lagrangiana:

$$L_i = \sum_{i=1}^n K_i - \sum_{i=1}^n P_i\tag{3.9}$$

Aplicando las Ecuaciones 3.7-3.9 y considerando lo observado en la Figura 3.3, se obtienen las siguientes ecuaciones dinámicas:

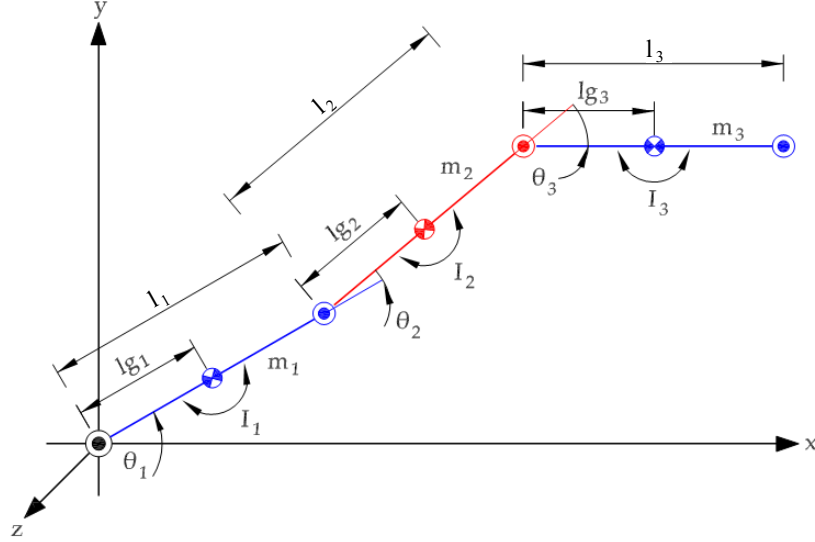


Figura 3.3: Parámetros de los enlaces para el modelo dinámico [9]

$$\begin{aligned}
 L &= K_1 + K_2 + K_3 - P_1 - P_2 - P_3 \\
 L &= \frac{1}{2}m_1 l_{g1}^2 \dot{\theta}_1^2 + \frac{1}{2}I_1 \dot{\theta}_1^2 + \frac{1}{2}m_2 \left[ l_1^2 \dot{\theta}_1^2 + l_{g2}^2 (\dot{\theta}_1 + \dot{\theta}_2)^2 + 2l_1 l_{g2} \dot{\theta}_1 (\dot{\theta}_1 + \dot{\theta}_2) C_2 \right] + \frac{1}{2}I_2 (\dot{\theta}_1 + \dot{\theta}_2)^2 \\
 &+ \frac{1}{2}m_3 [l_1^2 \dot{\theta}_1^2 + l_2^2 (\dot{\theta}_1 + \dot{\theta}_2)^2 + l_{g3}^2 (\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3)^2 + 2l_1 l_2 \dot{\theta}_1 (\dot{\theta}_1 + \dot{\theta}_2) C_2 \\
 &+ 2l_1 l_{g3} \dot{\theta}_1 (\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3) C_{23} + 2l_2 l_{g3} (\dot{\theta}_1 + \dot{\theta}_2) (\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3) C_3] + \frac{1}{2}I_3 (\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3)^2 \\
 &- m_1 g l_{g1} S_1 - m_2 g (l_1 S_1 + l_{g2} S_{12}) - m_3 g (l_1 S_1 + l_2 S_{12} + l_{g3} S_{123})
 \end{aligned} \tag{3.10}$$

El torque para la unión 1,  $\tau_1$ , se obtiene derivando la Ecuación 3.10 con respecto a  $\theta_1$ :

$$\begin{aligned}
 \frac{\partial L}{\partial \theta_1} &= m_1 l_{g1}^2 \theta_1 + I_1 \dot{\theta}_1 + m_2 [l_1^2 \dot{\theta}_1 + l_{g2}^2 (\dot{\theta}_1 + \dot{\theta}_2) + l_1 l_{g2} C_2 (2\dot{\theta}_1 + \dot{\theta}_2)] \\
 &+ I_2 (\dot{\theta}_1 + \dot{\theta}_2) + m_3 [l_1^2 \dot{\theta}_1 + l_2^2 (\dot{\theta}_1 + \dot{\theta}_2) + l_{g3}^2 (\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3) \\
 &+ l_1 l_2 C_2 (2\dot{\theta}_1 + \dot{\theta}_2) + l_1 l_{g3} C_{23} (2\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3) + l_2 l_{g3} C_3 (2\dot{\theta}_1 + 2\dot{\theta}_2 + \dot{\theta}_3)] \\
 &+ I_3 (\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3)
 \end{aligned} \tag{3.11}$$

La derivada en función del tiempo de la Ecuación 3.11 es:

$$\begin{aligned}
\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{\theta}_1} \right) = & \ddot{\theta}_1 [m_1 l_{g1}^2 + I_1 + m_2 (l_1^2 + l_{g2}^2 + 2l_1 l_{g2} C_2) + I_2 + m_3 (l_1^2 + l_2^2 + l_{g3}^2 + 2l_1 l_2 C_2 \\
& + 2l_1 l_{g3} C_{23} + 2l_2 l_{g3} C_3) + I_3] + \ddot{\theta}_2 [m_2 (l_{g2}^2 + l_1 l_{g2} C_2) + I_2 + m_3 (l_2^2 + l_{g3}^2 + l_1 l_2 C_2 \\
& + l_1 l_{g3} C_{23} + 2l_2 l_{g3} C_3) + I_3] + \ddot{\theta}_3 [m_3 (l_{g3}^2 + l_1 l_{g3} C_{23} + l_2 l_{g3} C_3) + I_3] \\
& - m_2 l_1 l_{g2} S_2 (2\dot{\theta}_1 \dot{\theta}_2 + \dot{\theta}_2^2) - m_3 l_1 l_2 S_2 (2\dot{\theta}_1 \dot{\theta}_2 + \dot{\theta}_2^2) - m_3 l_1 l_{g3} S_{23} (2\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3) (\dot{\theta}_2 + \dot{\theta}_3) \\
& - m_3 l_2 l_{g3} S_3 (2\dot{\theta}_1 \dot{\theta}_3 + 2\dot{\theta}_2 \dot{\theta}_3 + \dot{\theta}_3^2)
\end{aligned} \tag{3.12}$$

La derivada parcial de la función Langraniana es:

$$\frac{\partial L}{\partial \theta_1} = -m_1 g l_{g1} C_1 - m_2 g (l_1 C_1 + l_{g2} C_{12}) - m_3 g (l_1 C_1 + l_2 C_{12} + l_{g3} C_{123}) \tag{3.13}$$

Finalmente, se calcula  $\tau_1$ :

$$\begin{aligned}
\tau_1 = & \frac{d}{dt} \left( \frac{\partial L}{\partial \dot{\theta}_1} \right) - \frac{\partial L}{\partial \theta_1} \\
= & \ddot{\theta}_1 [m_1 l_{g1}^2 + I_1 + m_2 (l_1^2 + l_{g2}^2 + 2l_1 l_{g2} C_2) + I_2 + m_3 (l_1^2 + l_2^2 + l_{g3}^2 + 2l_1 l_2 C_2 \\
& + 2l_1 l_{g3} C_{23}) + I_3] + \ddot{\theta}_2 [m_2 (l_{g2}^2 + l_1 l_{g2} C_2) + I_2 + m_3 (l_2^2 + l_{g3}^2 + l_1 l_2 C_2 + l_1 l_{g3} C_{23} \\
& + 2l_2 l_{g3} C_3) + I_3] + \ddot{\theta}_3 [m_3 (l_{g3}^2 + l_1 l_{g3} C_{23} + l_2 l_{g3} C_3) + I_3] \\
& - m_2 l_1 l_{g2} S_2 (2\dot{\theta}_1 \dot{\theta}_2 + \dot{\theta}_2^2) - m_3 l_1 l_2 S_2 (2\dot{\theta}_1 \dot{\theta}_2 + \dot{\theta}_2^2) - m_3 l_1 l_{g3} S_{23} (2\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3) (\dot{\theta}_2 + \dot{\theta}_3) \\
& - m_3 l_2 l_{g3} S_3 (2\dot{\theta}_1 \dot{\theta}_3 + 2\dot{\theta}_2 \dot{\theta}_3 + \dot{\theta}_3^2) + m_1 g l_{g1} C_1 + m_2 g (l_1 C_1 + l_{g2} C_{12}) \\
& + m_3 g (l_1 C_1 + l_2 C_{12} + l_{g3} C_{123})
\end{aligned} \tag{3.14}$$

El mismo procedimiento es aplicado para las articulaciones 2 y 3, obteniendo:

$$\begin{aligned}
\tau_2 = & \frac{d}{dt} \left( \frac{\partial L}{\partial \dot{\theta}_2} \right) - \frac{\partial L}{\partial \theta_2} \\
= & \ddot{\theta}_1 [m_2 (l_{g2}^2 + l_1 l_{g2} C_2) + I_2 + m_3 (l_2^2 + l_{g3}^2 + l_1 l_2 C_2 + l_1 l_{g3} C_{23} + 2l_2 l_{g3} C_3) + I_3] \\
& + \ddot{\theta}_2 [m_2 l_{g2}^2 + I_2 + m_3 (l_2^2 + l_{g3}^2 + 2l_2 l_{g3} C_3) + I_3] + \ddot{\theta}_3 [m_3 (l_{g3}^2 + l_2 l_{g3} C_3) + I_3] \\
& - m_2 l_1 l_{g2} S_2 \dot{\theta}_1 \dot{\theta}_2 - m_3 l_1 l_2 S_2 \dot{\theta}_1 \dot{\theta}_2 - m_3 l_1 l_{g3} S_{23} \dot{\theta}_1 (\dot{\theta}_2 + \dot{\theta}_3) \\
& - m_3 l_2 l_{g3} S_3 (2\dot{\theta}_1 + 2\dot{\theta}_2 + \dot{\theta}_3) (\dot{\theta}_3) + m_2 l_1 l_{g2} S_2 \dot{\theta}_1 (\dot{\theta}_1 + \dot{\theta}_2) + m_3 l_1 l_2 S_2 \dot{\theta}_1 (\dot{\theta}_1 + \dot{\theta}_2) \\
& + m_3 l_1 l_{g3} S_{23} \dot{\theta}_1 (\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3) + m_2 g l_{g2} C_{12} + m_3 g (l_2 C_{12} + l_{g3} C_{123})
\end{aligned} \tag{3.15}$$

$$\tau_3 = \frac{d}{dt} \left( \frac{\partial L}{\partial \dot{\theta}_3} \right) - \frac{\partial L}{\partial \theta_3}$$

$$\begin{aligned} \tau_3 = & \ddot{\theta}_1 [m_3 (l_{g3}^2 + l_1 l_{g3} C_{23} + l_2 l_{g3} C_3) + I_3] + \ddot{\theta}_2 [m_3 (l_{g3}^2 + l_2 l_{g3} C_3) + I_3] + \ddot{\theta}_3 [m_3 l_{g3}^2 + I_3] \\ & + m_3 l_1 l_{g3} S_{23} \dot{\theta}_1^2 + m_3 l_2 l_{g3} S_3 (\dot{\theta}_1 + \dot{\theta}_2)^2 + m_3 g l_{g3} C_{123} \end{aligned} \quad (3.16)$$

### 3.3. Diseño mecánico

El exoesqueleto es un dispositivo usable con tres **DoF**, en el que la cadera, la rodilla y el tobillo son articulaciones activas. Los materiales que se emplean en la estructura son: el aluminio, para los eslabones que unen las articulaciones y para los soportes de los sistemas de reducción y, el acero para los engranajes y los ejes de transmisión de torque.

En el marco del prototipo, se han colocado cuatro abrazaderas que se ajustan con Velcro<sup>1</sup> a lo largo de la pierna. El desarrollo del exoesqueleto se ha centrado en el movimiento que realiza la extremidad en el plano sagital. Los eslabones que unen las articulaciones pueden ajustar su longitud mediante un mecanismo de dos barras telescópicas que se acoplan mediante tornillos.

Los actuadores son motores *brushless* **EC**<sup>2</sup>. Éstos han sido escogidos porque ofrecen mayores ventajas para dispositivos usables, pues tienen mayor densidad de torque, confiabilidad, son menos ruidosos, tienen mayor durabilidad y baja interferencia electromagnética. Los requerimientos de torque y velocidad angular en cada articulación, no pueden obtenerse directamente del actuador, por lo que se emplea un sistema de reducción mecánico acoplado al eje del motor.

#### 3.3.1. Reductores de velocidad

En la mayoría de los casos, un motor no tiene el torque necesario para poner en marcha un mecanismo, o las revoluciones que alcanza sobrepasan los requerimientos del sistema. Por este motivo, se debe utilizar un medio para disminuir las revoluciones e incrementar el torque hasta cierto nivel. A estos dispositivos se los conoce como sistemas de reducción [41].

##### 3.3.1.1. Reductor de la articulación de la cadera

Para esta articulación, se consideró un reductor cicloidal. Estos mecanismos son capaces de reducir la velocidad del eje de entrada y presentar a su salida relaciones relativamente altas en tamaños compactos. Consta de los cuatro componentes principales mostrados en la Figura 3.4:

- Eje de entrada,

---

<sup>1</sup>Compañía que ofrece cierres mecánicos de gancho y terciopelo

<sup>2</sup>Motor de **DC** que lleva incorporado una conmutación a **AC**



- Eje de salida,
- Disco cicloide, y,
- Carcasa con pasadores o pines internos.

Su funcionamiento se basa en un eje excéntrico que gira conjuntamente con el eje del motor en la entrada. Éste a su vez mueve un disco cicloidal cuyo perímetro se desplaza en un piñón estacionario formado por un conjunto de ejes o pines móviles que hacen contacto con el disco. La relación de salida depende de las dimensiones relativas del disco y la carcasa [42].

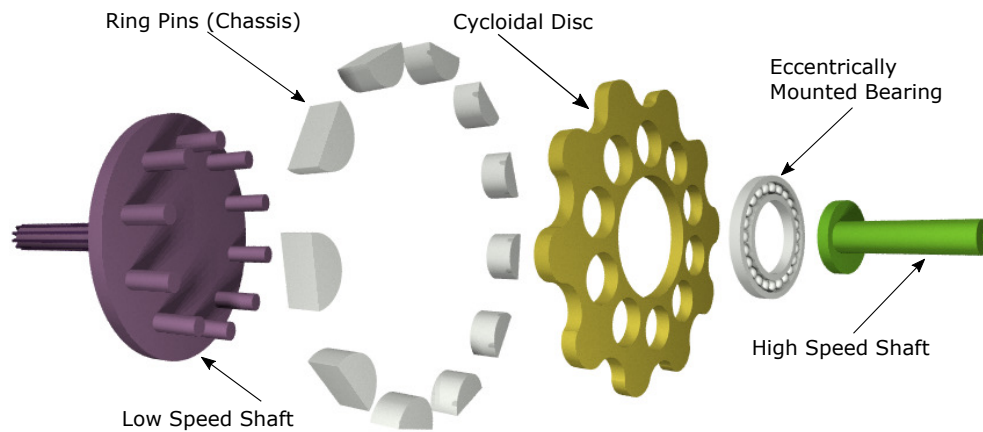
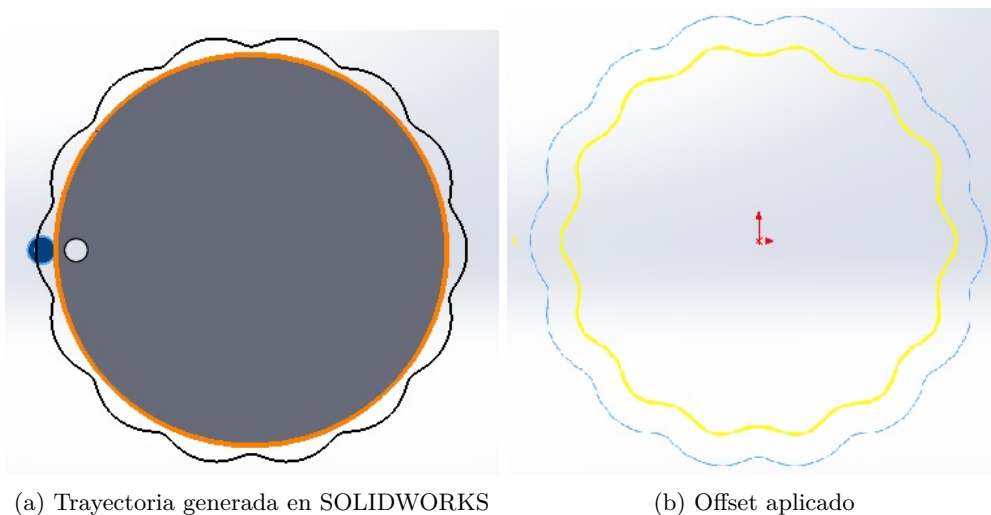


Figura 3.4: Elementos del reductor cicloidal [13]

El disco cicloide se obtuvo mediante un *offset* aplicado a una trayectoria generada en SOLIDWORKS [43], como se observa en la Figura 3.5.



(a) Trayectoria generada en SOLIDWORKS

(b) Offset aplicado

Figura 3.5: Diseño del disco cicloidal

Para el diseño del reductor, se considera un total de dos etapas colocadas en serie, de manera que la relación final sea el producto de las relaciones individuales. Cada etapa tiene una relación de 14:1, al tener las dos en serie se obtiene una relación de 196:1. Por lo tanto, el torque generado por el motor, será multiplicado 196 veces en el sistema, con una eficiencia del 87 % [44].

Los rodamientos utilizados en el prototipo, contemplan la masa de una persona de 75 Kg y se listan en la Tabla 3.2.

Tabla 3.2: Rodamientos empleados en el reductor cicloidal

Denominación	Carga dinámica $C$	Carga estática $C_0$	Ubicación
Rodillo de agujas SKF HK0609	2.81 kN	2.7 kN	Pasadores
Rodillo de agujas SKF HK0608	2.1 kN	1.8 kN	Eje de salida
Rodamiento rígido de bolas SKF 61900	2.7 kN	1.27 kN	Eje de entrada
Rodamiento de contacto angular SKF 7204	13.3 kN	7.65 kN	Soporte eje de salida
Rodamiento rígido de bolas Koyo 696z	1.35 kN	0.63 kN	Rodillo salida - entrada entre 1ra y 2da etapa
Rodamiento axial de rodillos SKF AXK5578	34.5 kN	186 kN	Soporte eje de salida

### 3.3.1.2. Reductor de las articulaciones de rodilla y tobillo

Para obtener una relación de transmisión elevada, se elige la construcción de un sistema de engranajes epicicloidal o planetario. Este tipo de mecanismo consta de un engranaje central denominado sol, en torno al cual gira un trío de engranajes denominados satélites o planetas sujetos mediante un porta-satélites. Este conjunto gira dentro de un engranaje exterior conocido como corona. La Figura 3.6 muestra el sistema planetario y sus componentes.

El diseño del sistema de engranajes, requiere de algunos cálculos adicionales para determinar el número correcto de dientes en cada elemento, por conveniencia se sigue la siguiente nomenclatura:

$R$ : número de dientes del engranaje corona.

$S$ : número de dientes del engranaje sol.

$P$ : número de dientes de los engranajes planeta.

Para todo sistema planetario, se debe cumplir que el módulo sea igual para todos los engranajes, también se debe cumplir la Ecuación 3.17:

$$R = 2P + S \quad (3.17)$$

Con  $S = 22$  y  $P = 50$ , se obtiene el número de dientes para la corona  $R = 122$ . La relación de transmisión del sistema se obtiene de la Ecuación 3.18:

$$(R + S) \cdot T_y = R \cdot T_r + T_s \cdot S \quad (3.18)$$

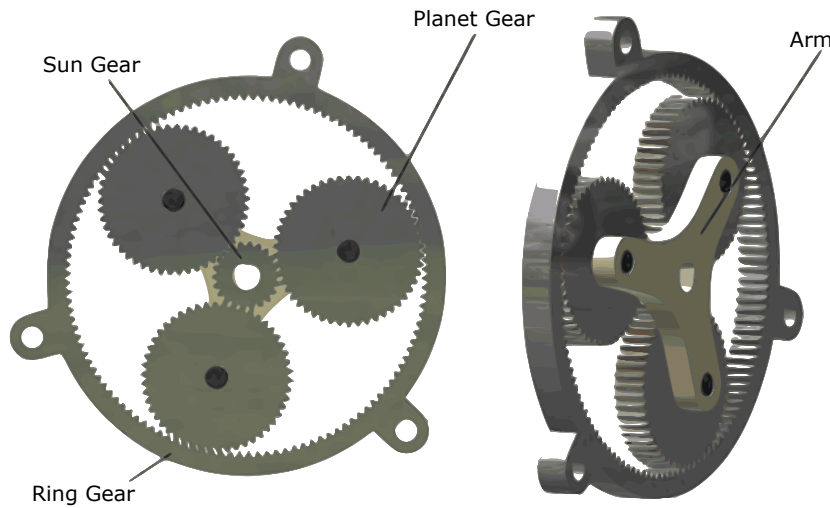


Figura 3.6: Elementos de un sistema de reducción planetario

donde  $T_r$ ,  $T_s$  y  $T_y$ , es la velocidad angular de los engranajes corona, sol y el portasatélites respectivamente. El elemento fijo en este caso es la corona, por lo que  $T_r = 0$ . La entrada es el engranaje sol y la salida es el portasatélites. Su velocidad está dada por:

$$T_y = T_s \frac{S}{R + S} \quad (3.19)$$

Reemplazando los valores de  $R$ , y  $S$  se obtiene una relación de transmisión  $= 0.152$ , es decir, si el engranaje sol gira  $360^\circ$ , el portasatélites realizará un giro de  $55^\circ$  aproximadamente. Por lo tanto, serán necesarias 6.545 vueltas del engranaje sol para completar una vuelta del portasatélites. Cabe recalcar que se emplearán dos sistemas de engranajes planetarios en serie,

obteniendo una relación de 0.0234 o 1 a 42.8, por lo que una vuelta del manipulador corresponde a 42.8 vueltas del motor.

El peso y volumen que ocupará el sistema son muy importantes, por lo que el módulo del engranaje es igual a 0.5, con ello el diámetro exterior del conjunto armado es de 68 mm. Los elementos se generan en SOLIDWORKS y las dimensiones obtenidas se listan en la Tabla 3.3.

Tabla 3.3: Descripción de los engranajes

Engranaje	Diámetro primitivo [mm]	Espesor [mm]
Sol	11	4
Planeta	25	4
Corona	61	4

### 3.3.1.3. Análisis de materiales

SOLIDWORKS, ofrece una herramienta de simulación que permite estimar el comportamiento de un elemento en su entorno de funcionamiento. En el caso de los engranajes, se requiere conocer cual es el límite de presión que pueden soportar sus dientes, ya que éstos se encargan de transmitir el torque generado por el actuador.

Para la simulación se asigna una aleación de acero al carbono a un engranaje aislado, se ancla el eje a una geometría fija y se aplica una fuerza a uno de los dientes, incrementando su valor hasta obtener el límite elástico para el elemento. La Figura 3.7, muestra los resultados de la simulación efectuados en el engranaje sol, que es el elemento más pequeño del conjunto. El material con el que se trabaja es una aleación de acero al carbono conocida como k100<sup>3</sup>, puesto que ofrece un excelente acabado, resistencia a la corrosión y al desgaste.

---

<sup>3</sup>Norma DIN 1.2080

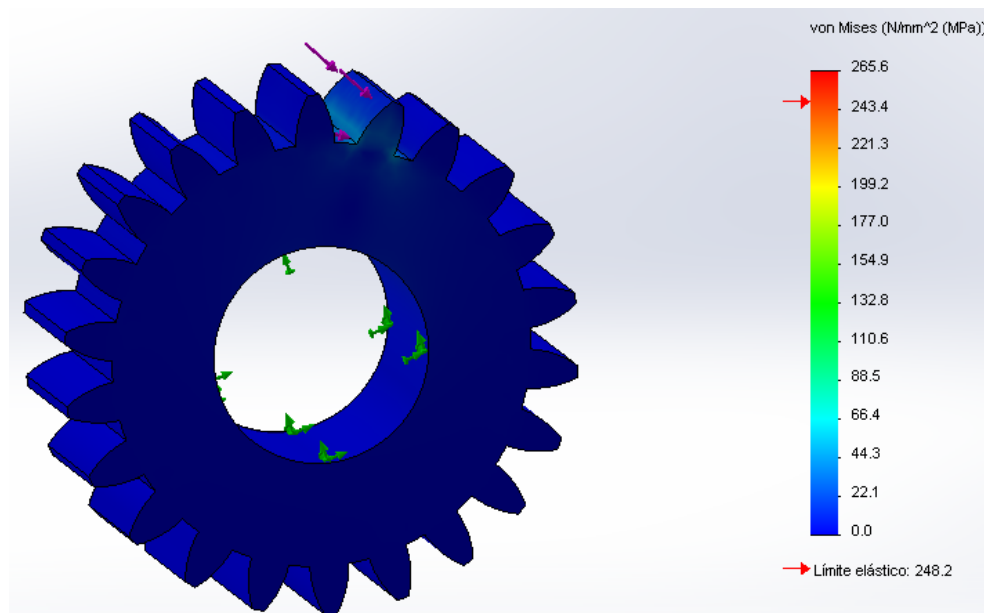


Figura 3.7: Resultados de la simulación en SOLIDWORKS

Los rodamientos que forman parte de este sistema se especifican en la Tabla 3.4. Se considera un sujeto con una masa de 75 Kg.

Tabla 3.4: Rodamientos empleados en el reductor planetario

Denominación	Carga dinámica $C$	Carga estática $C_0$	Ubicación
Rodillo de agujas SKF 6001	5.4 kN	2.36 kN	Soporte eje de salida
Rodillo de bolas SKF 625	1.14 kN	0.38 kN	Eje de entrada

Los planos del sistema reductor se detallan el Apéndice F.

### 3.4. Diseño 3D

Parte del trabajo fue dejar un prototipo funcional y estético, logrando cierto profesionalismo en lo realizado. En la parte estética se diseñó una carcasa para cubrir a todos los elementos de la articulación, incluyendo al motor, *driver* y su respectivo cableado. La carcasa consta de dos piezas: la base, que va asegurada en la estructura mecánica; y el armazón, que se sujeta en la base para cubrir a la articulación robótica.

Esta carcasa fue diseñada en SOLIDWORKS y creada en una impresora 3D Robo R2 [45] a base de material de ácido poliláctico (PLA, por sus siglas en inglés) de 1.75 mm de diámetro. En el diseño se consideró una forma genérica para que las carcasas puedan ser intercambiables entre articulaciones sin ningún problema.

Por otra parte, la articulación del tobillo funciona con un *encoder* externo, por lo que se implementó un acople para el motor, el cual, consta de dos partes: un acople móvil, el mismo que transmite el movimiento del motor a través de un eje para que pueda ser leído por el *encoder*; y un acople fijo, que da soporte y estabilidad al *encoder*.

Acorde a un análisis subjetivo y empírico sobre este tipo de impresiones 3D con material PLA, se pudo observar que los objetos tienen una gran resistividad a pesar de que los mismos fueron impresos con un espesor de dos milímetros y una calidad estándar. No se realizaron pruebas de resistividad y calidad debido al tiempo que oscilaba entre 12 horas para una calidad estándar y 40 horas para una calidad alta.

Los planos de las piezas 3D se detallan en el Apéndice E.

## Capítulo 4

# Desarrollo del sistema de control, comunicación e instrumentación

### 4.1. Introducción

Este capítulo presenta el desarrollo de los sistemas de control y de comunicaciones entre los 3 controladores ([EPOS4](#)) y una Raspberry Pi 3, que permiten realizar el movimiento de tres articulaciones (Cadera, Rodilla y Tobillo) en un exoesqueleto de extremidades inferiores para la pierna izquierda. Dicho exoesqueleto será utilizado para la rehabilitación de pacientes que sufran de alguna incapacidad en sus piernas. Las articulaciones del exoesqueleto deben imitar el movimiento de una pierna humana, es decir, deben seguir la trayectoria de cada una de las articulaciones durante la marcha. Para ello, es necesario colocar un motor en cada una de las articulaciones, con su respectivo controlador ([EPOS4](#)). Posteriormente se deben ingresar tres señales de referencia (trayectorias) que los motores deben seguir, una para cada articulación. Las señales de referencia se ingresan en una Raspberry Pi 3 que está encargada de almacenar las referencias y enviarlas a los tres [EPOS4](#) para que se genere el movimiento. La referencia ingresada en la Raspberry Pi 3 está conformada por los puntos máximos y mínimos de la curva de referencia original, para que los controladores realicen una interpolación con esos valores y se forme la curva de la referencia deseada. La Raspberry Pi 3 se programa en C++ y mediante la

librería *EPOSCmd*, propia del [EPOS4](#), se puede seleccionar el tipo de control para los motores, la comunicación entre los tres controladores y las posiciones que deben tener los motores de acuerdo a la referencia antes mencionada.

Se aplica una comunicación mediante un bus [CAN](#) entre los controladores [EPOS4](#) de cada articulación y la Raspberry Pi 3. Mediante la comunicación se logra enviar información desde la Raspberry Pi 3 a cada uno de los [EPOS4](#) del exoesqueleto y también se lee información en la Raspberry Pi 3. Todos los mensajes enviados por el bus [CAN](#) están formados como tramas [SDO](#), que como se mencionó anteriormente, son tramas propias del protocolo CANopen.

Los [EPOS4](#) utilizan el modo "perfil de operación" (*Profile Position Mode*) el cual utiliza un control [PID](#) con una interpolación parabólica. Así, los motores de cada articulación generan la trayectoria de la referencia original. Los controladores, además de realizar el control y la comunicación, permiten que los motores se muevan entregándoles la corriente y voltaje necesarios. Finalmente, desde el [EPOS4](#) se obtiene información de la corriente consumida, la velocidad de los motores, la posición y torque de cada una de las articulaciones. El diagrama de bloques que representa el funcionamiento del sistema de control y comunicación para el movimiento del exoesqueleto de extremidades inferiores se presenta en la Figura 4.1.

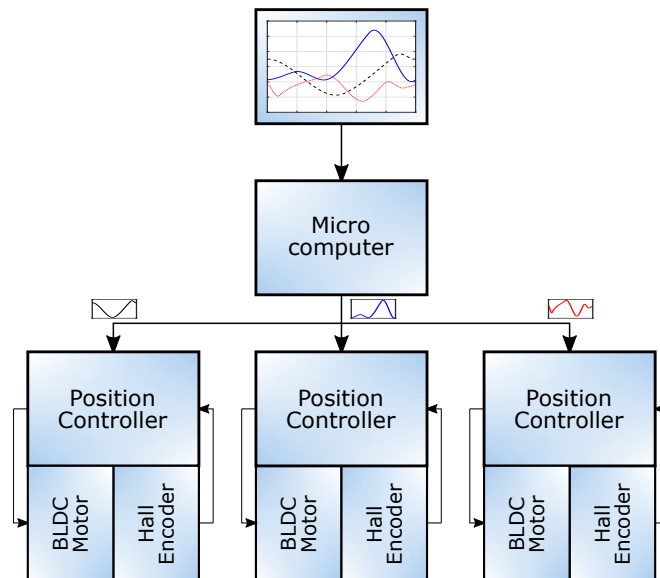


Figura 4.1: Diagrama de bloques implementado

Este capítulo está organizado de la siguiente manera: la Sección 4.2 presenta la obtención de las referencias de cada una de las articulaciones; la Sección 4.3 describe la conexión entre los motores, los drivers, el ajuste de las constantes de control y la topología de red utilizada en el exoesqueleto; la Sección 4.4 trata sobre las transformaciones de unidades necesarias para lograr un movimiento preciso en las articulaciones y la interpolación parabólica implementada



en los [EPOS4](#); finalmente la Sección [4.5](#) indica cómo se realiza la programación dentro de la Raspberry Pi 3 para controlar y tener una comunicación directa con cada una de las articulaciones del exoesqueleto, además explica las funciones que se utilizan para realizar la captura de la información relevante del exoesqueleto.

## 4.2. Señales de referencia

En [\[46\]](#) se explica cómo actúa la biomecánica angular de las articulaciones de la cadera, rodilla y tobillo durante un ciclo de caminata. En las Tablas [4.1](#), [4.2](#) y [4.3](#) se presenta la posición angular de cada articulación en su determinado porcentaje del ciclo de marcha.

El movimiento de cada una de las articulaciones debe ser suavizado para llegar a cada uno de los ángulos indicados. Para ello es necesario generar una trayectoria que cubra los ángulos de cada articulación mediante interpolación. En la Figura [4.2](#) se muestra la trayectoria de cada una de las articulaciones durante un ciclo completo de marcha. En la Sección [4.4.2](#) se explica la interpolación utilizada en este proyecto para generar las trayectorias de cada articulación a partir de las variaciones angulares.

Tabla 4.1: Referencia de la cadera

Posición (grados)	Porcentaje de la caminata (%)
30	0
-15	52
35	87
30	100

Tabla 4.2: Referencia de la rodilla

Posición (grados)	Porcentaje de la caminata (%)
3	0
13	18
3	40
65	69
3	100

Tabla 4.3: Referencia del tobillo

Posición (grados)	Porcentaje de la caminata (%)
-4	0
-18	8
8	38
-24	67
-2	79
-8	92
-4	100

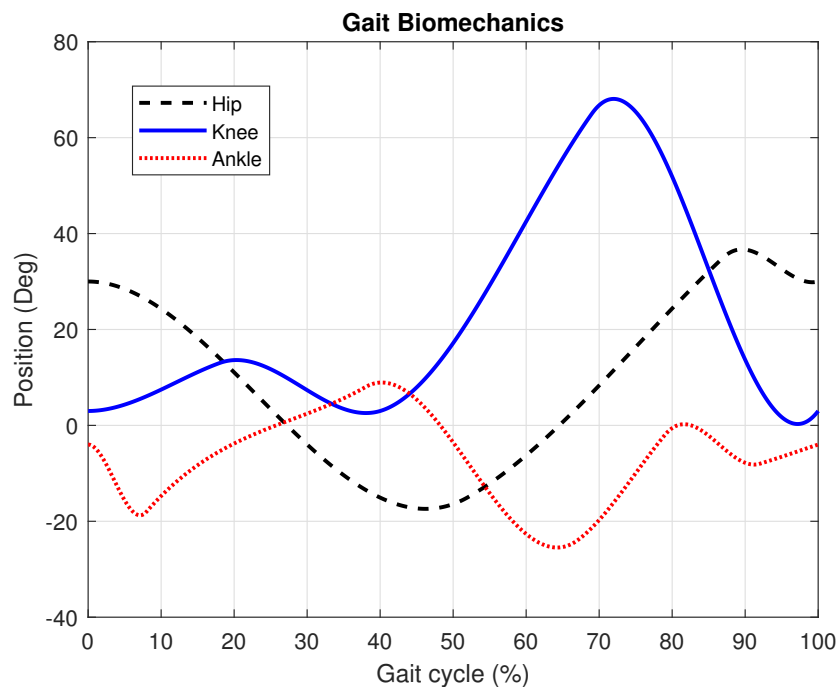


Figura 4.2: Trayectorias de las tres articulaciones durante un ciclo de marcha

### 4.3. Conexiones y configuraciones entre motores y EPOS4 de cada articulación

Cada una de las articulaciones del exoesqueleto está formada por un motor (Cadera: *Maxon EC90 flat 90w* y reducción de 196:1, Rodilla y Tobillo: *Maxon EC45 flat 70w* con reducción de 42.8:1) y un controlador [EPOS4](#). Tanto los motores como los controladores son del fabricante

*Maxon Motor*, el mismo que facilita los diagramas de conexión entre los dos dispositivos y las configuraciones necesarias para realizar el control.

Los **EPOS4** son dispositivos esclavos que están comandados por una Raspberry Pi 3, que es el dispositivo maestro en la comunicación. La Raspberry Pi 3 utiliza una tarjeta PiCANv2 [47] para establecer una comunicación **CAN** completa entre el dispositivo maestro y los esclavos colocados en cada articulación. Toda la red utiliza el protocolo CANopen, con el cual, la Raspberry Pi 3 puede enviar y recibir información a cada una de las articulaciones. Cada **EPOS4** tiene un identificador, que le permite a la Raspberry Pi 3 conocer a qué articulación pertenece. Con la ayuda del identificador, la Raspberry Pi 3 sabrá a qué articulación debe enviar y/o recibir información.

#### 4.3.1. Conexiones iniciales entre los motores y los EPOS4

Para utilizar los motores colocados en cada una de las articulaciones, es necesario realizar una correcta conexión hacia su respectivo controlador **EPOS4**. Los motores únicamente podrán funcionar con los controladores, ya que estos dispositivos brindan la corriente y voltaje necesarios para su funcionamiento. Además, los **EPOS4** implementan estrategias de control que permiten realizar el movimiento de los motores según la trayectoria deseada.

La información necesaria para la conexión de los motores se encuentra en las hojas de especificaciones [48], [49], [50], [51] de los fabricantes (*Maxon Motor*). Las conexiones necesarias entre los motores y los **EPOS4** se detallan en el Apéndice C. Se debe tener cuidado al momento de conectar los dispositivos, ya que una mala conexión puede ocasionar daños permanentes en los motores o controladores.

En este proyecto se diseñaron placas de circuito impreso (**PCB**, por sus siglas en inglés), para cada articulación. Las placas tienen la finalidad de conectar los **EPOS4**, los motores y cables de comunicación en un sólo lugar de manera sencilla, para evitar conexiones incorrectas entre los dispositivos. El diseño de los **PCBs** se lo realiza en el *software Altium Designer*. El diseño esquemático y las placas finales se presentan en el Apéndice C.

#### 4.3.2. Configuración de los EPOS4

Los dispositivos **EPOS4** necesitan ser configurados antes de utilizarlos en cualquier aplicación. Dentro de la configuración se debe ingresar el identificador, velocidad de transmisión para la comunicación **CAN** y los parámetros del motor que va a controlar cada **EPOS4**. La configuración de los módulos **EPOS4** se lo realiza en el *software EPOS Studio* del fabricante (compatible únicamente con el sistema operativo *Windows*).

Dentro de *EPOS Studio*, se debe crear un proyecto nuevo y conectar el **EPOS4** de cada articulación de manera individual mediante un cable **USB** hacia un computador para configurarlos. Lo primero será colocar el identificador a cada **EPOS4** y la velocidad de transmisión, como se puede ver en la Figura 4.3. El **EPOS4** de la cadera tendrá el identificador *Node 1*, el de la rodilla el *Node 2* y el del tobillo el *Node 3*. Todos los dispositivos tendrán una velocidad de 1 **Mbps** para transmitir en el bus **CAN**.

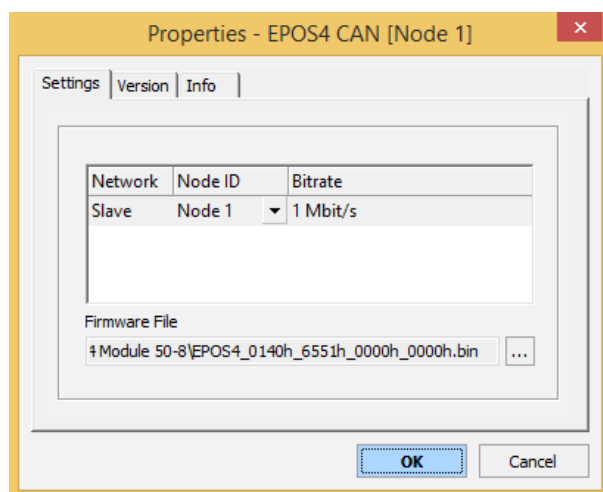


Figura 4.3: Identificador y *bitrate* del **EPOS4** colocado en la cadera

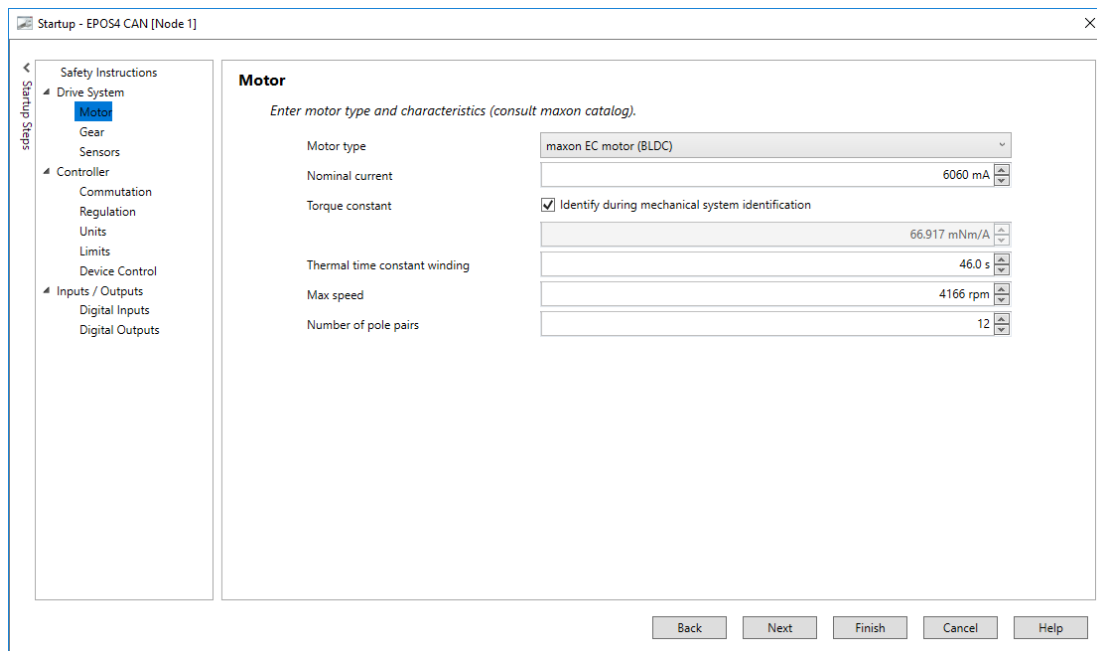
Posteriormente se deben ingresar los parámetros del motor a comandar. Dentro de *EPOS Studio* existe una herramienta llamada *Startup Wizard* que permite ingresar los parámetros de una manera sencilla. Los parámetros que se requieren ingresar en la configuración del **EPOS4** se obtienen de las especificaciones técnicas de los motores (Apéndice B). Se ingresan los parámetros como: el tipo de motor, la corriente nominal, la constante de tiempo térmico (*termal time constant*) del devanado y el número de pares de polos (Figura 4.4).

Se configuran los sensores disponibles dentro del motor (*hall*, *encoder* interno y externo). Es importante colocar el número correcto de la resolución de los *encoders* utilizados. En este proyecto se utiliza una resolución de 1024 cpt (Figura 4.5).

Se selecciona el tipo de conmutación y sensores que utiliza el motor. Para los motores utilizados en este proyecto se necesita colocar la conmutación sinusoidal y seleccionar tanto los sensores *hall* como el *encoder* (Figura 4.6).

Se selecciona el tipo de control para la corriente, velocidad y posición que actuará sobre el motor. Además, se selecciona el sensor principal para realizar los controles mencionados. En este proyecto se selecciona al *encoder* como sensor principal (Figura 4.7).

Finalmente se ingresan los límites de operación, considerando que la máxima corriente de salida tiene que ser tres veces la corriente continua máxima. Además, es necesario colocar los



**Startup - EPOS4 CAN [Node 1]**

**Motor**

Enter motor type and characteristics (consult maxon catalog).

Motor type: maxon EC motor (BLDC)

Nominal current: 6060 mA

Torque constant: ☒ Identify during mechanical system identification

66.917 mNm/A

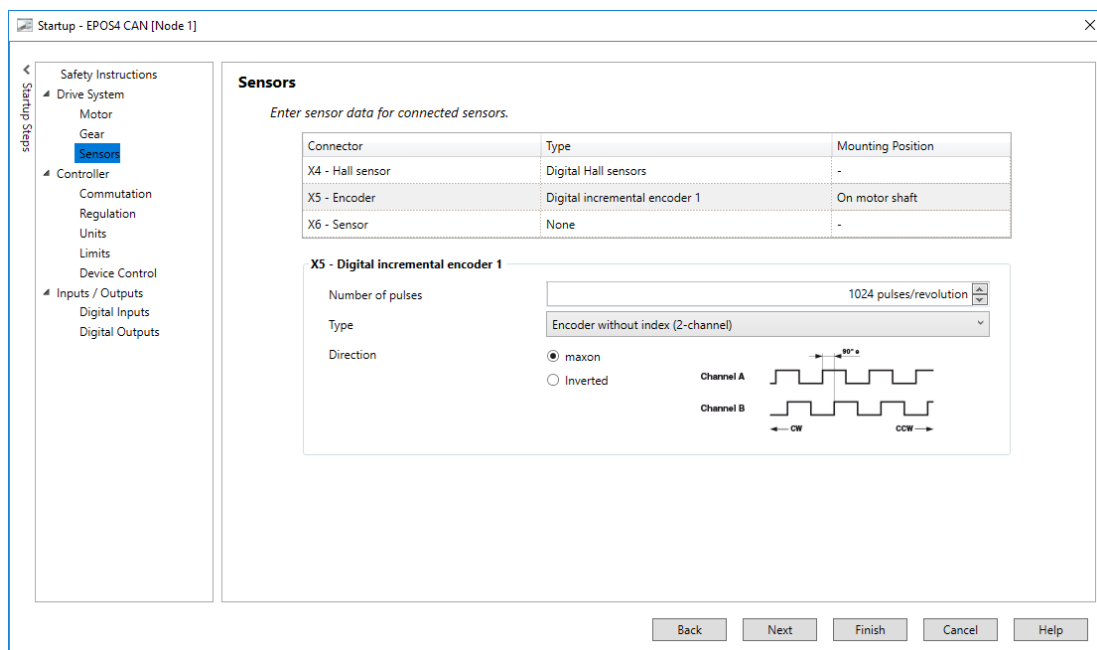
Thermal time constant winding: 46.0 s

Max speed: 4166 rpm

Number of pole pairs: 12

Back Next Finish Cancel Help

Figura 4.4: Ingreso de parámetros iniciales del motor a utilizar



**Startup - EPOS4 CAN [Node 1]**

**Sensors**

Enter sensor data for connected sensors.


Connector	Type	Mounting Position
X4 - Hall sensor	Digital Hall sensors	-
X5 - Encoder	Digital incremental encoder 1	On motor shaft
X6 - Sensor	None	-


**X5 - Digital incremental encoder 1**

Number of pulses: 1024 pulses/revolution

Type: Encoder without index (2-channel)

Direction: ☒ maxon ☐ Inverted

Channel A: 

Channel B: 

90°

CW CCW

Back Next Finish Cancel Help

Figura 4.5: Selección de sensores a utilizar

límites de las revoluciones de cada motor (Figura 4.8). Esta información se encuentra en las especificaciones técnicas de los motores, detalladas en el Apéndice B.

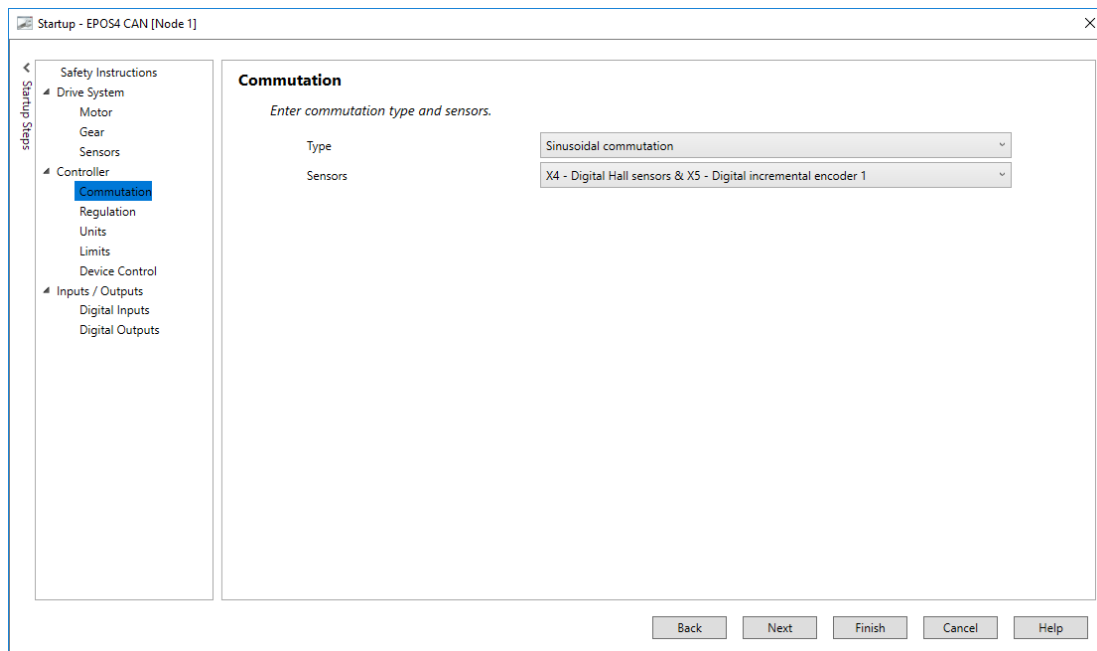


Figura 4.6: Seleccionar el tipo de conmutación y el sensor

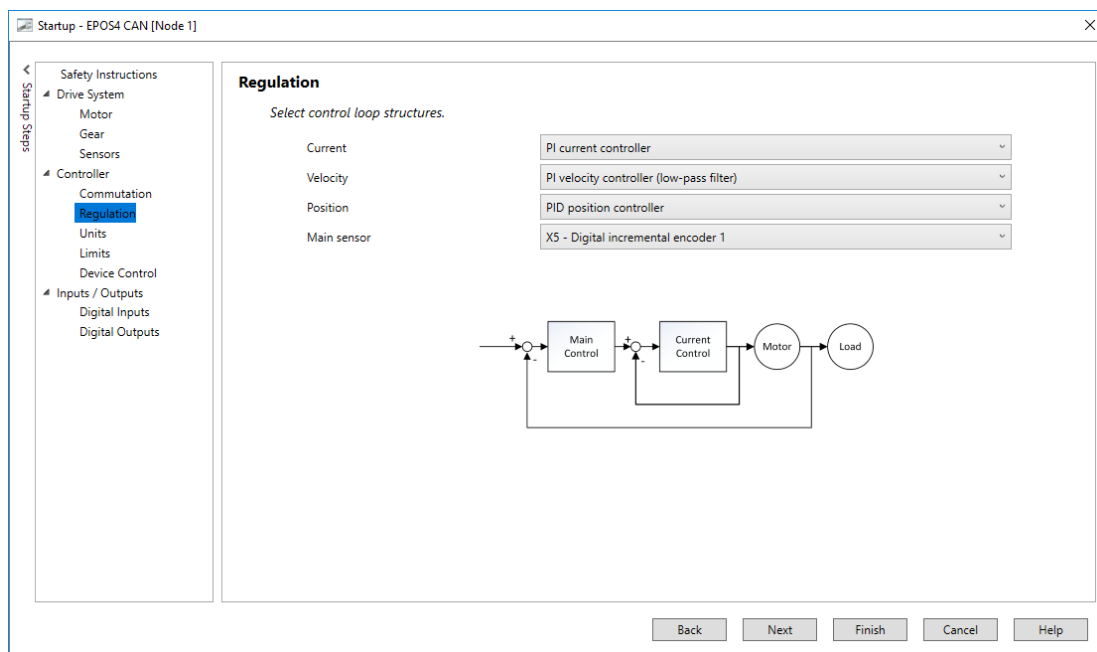


Figura 4.7: Selección de tipo de control y sensor principal en el EPOS4

La configuración debe realizarse en los EPOS4 de cada articulación una sola vez, y la información se almacena de manera permanente en cada EPOS4 hasta una siguiente reconfiguración.

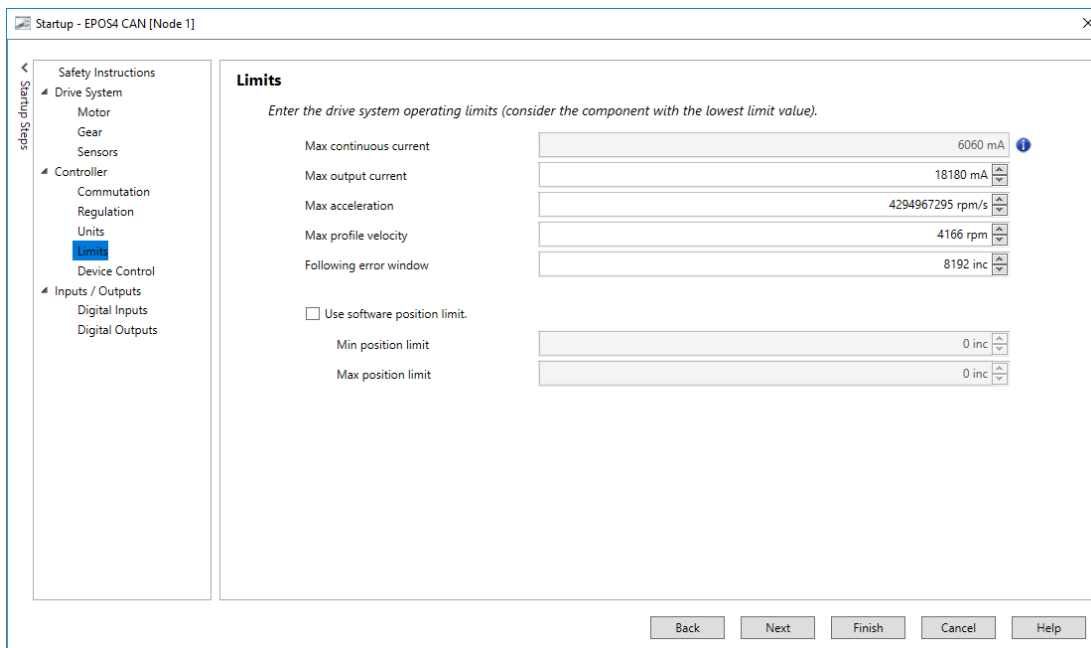


Figura 4.8: Selección de los límites de operación del EPOS4

Es decir, que cada vez que se utilicen los EPOS4 en cualquier aplicación no será necesario volver a utilizar *EPOS Studio* para realizar la configuración inicial. En el caso de que se cambie algún motor de una articulación, nuevamente será necesario realizar la configuración inicial con los parámetros de este nuevo motor.

#### 4.3.3. Ajuste de constantes para el control PID

Los controladores EPOS4 tienen varios modos de operación, y cada uno de ellos utiliza un control específico. Se debe seleccionar un modo de operación que permita a los motores de cada una de las articulaciones del exoesqueleto seguir las referencias indicadas en la Sección 4.2. Para este proyecto se utiliza el modo de operación *Profile Position Mode*, que utiliza el control PID implementado dentro de cada EPOS4. Además, utiliza una interpolación parabólica, que se explica en la Sección 4.4.2. Este modo de operación ayuda a tener un control de posición en cada uno de los motores. Únicamente se ingresa un valor de posición de referencia, y el control PID, conjuntamente con la interpolación parabólica, genera el movimiento de los motores de una manera suavizada, hasta llegar a la referencia.

Para el control PID que implementa el EPOS4, es necesario determinar las constantes P, I y D para que el control sea preciso al seguir las trayectorias de referencia requeridas, y replicar así un patrón de marcha natural. El ajuste de las constantes es muy sencillo, ya que dentro de *EPOS Studio* existe una herramienta que ayuda a realizar un ajuste automático de las constantes

para el control de posición. Para realizar el ajuste mencionado, es necesario tener acoplados los motores en la estructura del exoesqueleto para que las constantes sean calculadas con respecto a la fricción real a la que van a estar sujetas.

La herramienta para el ajuste de las constantes de control se llama sintonización de regulador (*Regulation Tuning*) (Figura 4.9), se selecciona el control de posición y una respuesta al impulso. Posteriormente se presiona el botón de sintonización automática (*auto tuning*). El motor comienza a funcionar girando en los dos sentidos para reconocer el sistema al que va a ser expuesto. Se debe esperar hasta que el motor se detenga, esto indica que el ajuste ha terminado y automáticamente se han calculado las constantes para el control PID. Una vez seleccionadas las constantes se puede aplicar una señal de referencia para observar el comportamiento del controlador y, si se desea, es posible modificar las constantes calculadas para realizar un ajuste específico en el control. Se debe repetir el mismo procedimiento para los motores de cada articulación. Para la aplicación del exoesqueleto se obtuvieron buenos resultados con las constantes calculadas automáticamente por lo que no fue necesario modificar las constantes de manera manual.

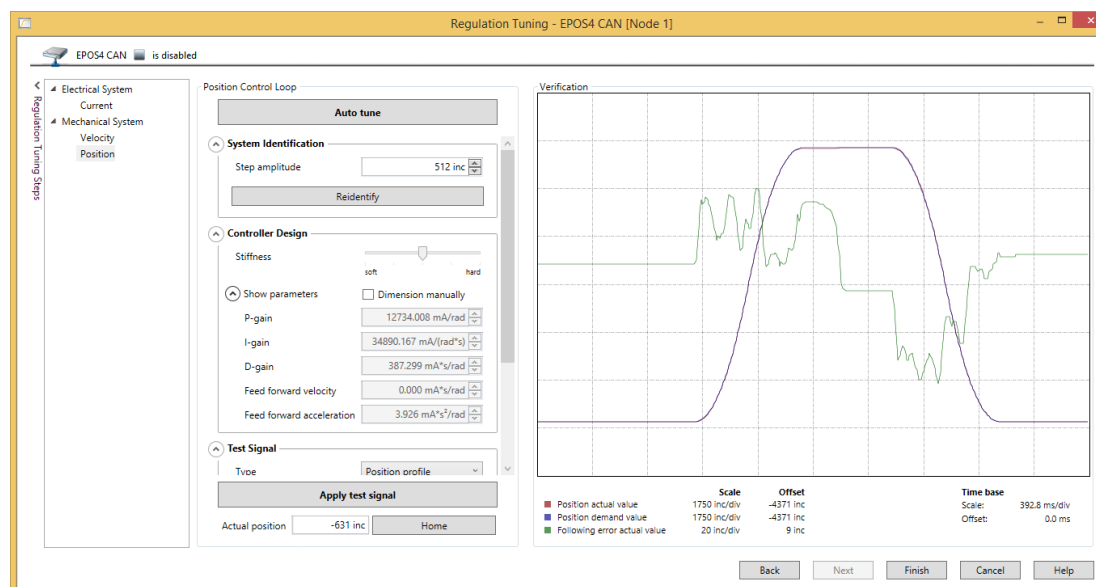


Figura 4.9: Ajuste de constantes y prueba del control PID

#### 4.3.4. Topología de red implementada en el exoesqueleto para la comunicación

Una vez conectado, configurado y montado los motores y EPOS4 en la estructura de exoesqueleto, se necesita crear una red de comunicaciones para intercambiar mensajes entre cada



uno de los [EPOS4](#) y un dispositivo maestro central, encargado de iniciar el movimiento del exoesqueleto y capturar información relevante de cada una de las articulaciones.

Se utiliza una red tipo bus [CAN](#), a la cual está conectada el dispositivo maestro (Raspberry Pi 3) y los dispositivos esclavos ([EPOS4](#)). Para tener un puerto [CAN](#) dentro del dispositivo maestro, se necesita un módulo extra llamado PiCANv2 compatible con los puertos [CAN](#) de los [EPOS4](#). La topología de red implementada para la comunicación se presenta en la Figura 4.10. Dentro de la arquitectura propuesta, la Raspberry Pi 3 es la encargada de enviar la información del movimiento hacia cada uno de los motores de las articulaciones del exoesqueleto y recibir información de las mismas como: posición, velocidad y corriente.

Los dispositivos esclavos son colocados en las articulaciones de la cadera, rodilla y tobillo respectivamente, que a su vez están conectados a un motor. Cada [EPOS4](#) permite controlar su respectivo motor acorde a los datos enviados por la Raspberry Pi 3. Los datos enviados sobre el bus [CAN](#) utilizan el protocolo CANopen.

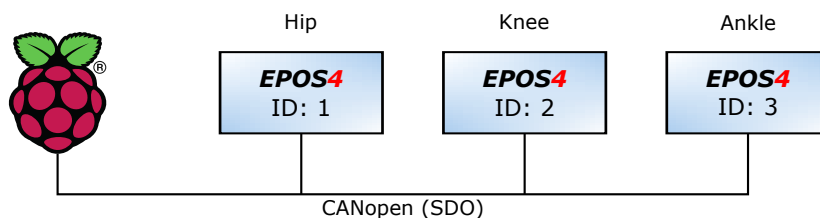


Figura 4.10: Topología de red utilizada en el exoesqueleto

Como ha sido mencionado, todos los [EPOS4](#) necesitan un identificador colocado en la configuración inicial. Con el identificador es posible conocer a qué articulación se direcciona el envío o recepción de un mensaje dentro de la red de comunicación. Las articulaciones de la cadera, rodilla y tobillo tienen los identificadores 1, 2 y 3 respectivamente. El identificador es necesario en los mensajes [SDO](#), ya que dentro de la trama enviada se debe ingresar el nodo al cual se desea enviar el mensaje, caso contrario el mensaje no tendría destinatario y se descarta el mensaje. Además, la velocidad de comunicación configurada en cada uno de los equipos de la red es de 1 [Mbps](#), la cual es la máxima tasa que se puede utilizar en un bus [CAN](#). Esto debido a que se requiere una comunicación inmediata entre todos los equipos conectados a la red.

## 4.4. Determinación de los parámetros del sistema

Es necesario determinar los parámetros que el exoesqueleto utiliza para generar su movimiento en valores que este lo entienda. Para ello se realiza una transformación de unidades que permita llevar los valores de posición presentados en la Sección 4.2 a unidades que los motores y la estructura del exoesqueleto admitan. Adicionalmente, se deben calcular parámetros como

velocidad y aceleración de cada uno de los motores y para cada uno de los tramos de la trayectoria de marcha de cada articulación. Los cálculos son necesarios en la generación de las trayectorias, ya que ayudan a mover a los motores de manera suavizada, lo que es indispensable en el movimiento del exoesqueleto.

#### 4.4.1. Transformaciones de unidades

Para un mejor entendimiento del operador y/o usuario del exoesqueleto, se utilizarán grados para definir la posición de las articulaciones y segundos para el tiempo que dura cada tramo de una caminata, sin embargo, los motores no son capaces de manejar directamente estas unidades.

Los motores utilizados en este proyecto son de la marca *Maxon Motor* y la unidad con la que definen la posición es el incremento. Los motores tienen incorporado un *encoder* que divide una revolución completa del motor en 4096 incrementos. Adicionalmente, en las unidades se deben considerar las reducciones que tiene el exoesqueleto acoplado a los motores de cada una de sus articulaciones. Por lo tanto, es necesario definir una relación de transformación que permita pasar de grados a incrementos considerando las reducciones de cada una de las articulaciones del exoesqueleto.

- **Cadera:** La caja reductora instalada en la articulación de cadera provoca que una revolución de la articulación sean 196 revoluciones en su motor (Figura 4.11).

$$\begin{aligned}1 \text{ rev} &= 196 \text{ rev}_{\text{motor}} \\1 \text{ rev}_{\text{motor}} &= 4096 \text{ incrementos} \\1 \text{ rev}_{\text{cadera}} &= 360^\circ\end{aligned}$$

Con las relaciones se obtiene la Ecuación 4.1 que permite transformar los datos de ingreso a unidades que maneja la articulación de la cadera, donde  $\alpha$  es el ángulo deseado en la articulación de la cadera, y  $\beta$  es el incremento necesario en el motor.

$$\beta = \alpha \frac{4096 \cdot 196}{360} \approx \alpha \cdot 2230 \quad (4.1)$$

- **Rodilla y tobillo:** Las cajas reductoras instaladas en las articulaciones de rodilla y tobillo provocan que una revolución de la articulación sean 42.8 revoluciones en su motor (Figura 4.12).

$$\begin{aligned}1 \text{ rev} &= 42.8 \text{ rev}_{\text{motor}} \\1 \text{ rev}_{\text{motor}} &= 4096 \text{ incrementos} \\1 \text{ rev}_{\text{rodilla/tobillo}} &= 360^\circ\end{aligned}$$

Con las relaciones se obtiene la Ecuación 4.2 que permite transformar los datos de ingreso a unidades que maneja la articulación de la rodilla y tobillo, donde  $\alpha$  es el ángulo deseado en la articulación de la rodilla y tobillo, y  $\beta$  es el incremento necesario en el motor.

$$\beta = \alpha \frac{4096 \cdot 42.8}{360} \approx \alpha \cdot 487 \quad (4.2)$$

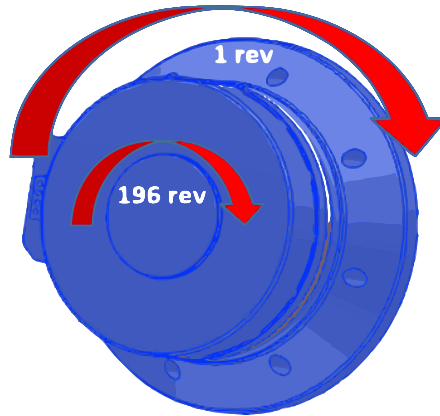


Figura 4.11: Relación de revoluciones en la cadera

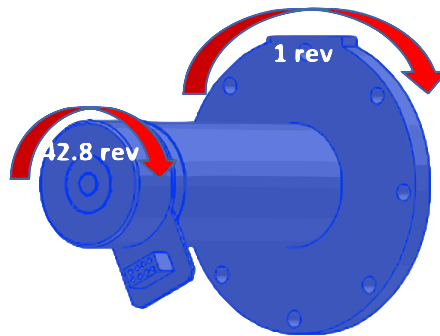


Figura 4.12: Relación de revoluciones en la rodilla y tobillo

Para mover los motores en el tiempo de caminata deseado, se requiere calcular la velocidad que deben tener los mismos en cada uno de los tramos de la caminata. Los motores únicamente trabajan en [rpm](#), por lo que es necesario realizar una nueva transformación de unidades.

Una vez obtenidas todas las posiciones de la referencia en incrementos con las transformaciones anteriores, se calcula el tiempo que se va a requerir para mover el motor durante cada tramo de la caminata. Para ello se requiere ingresar el tiempo que durará un ciclo de caminata y multiplicar por el porcentaje de caminata de cada tramo de la marcha (Sección [4.2](#)).

Con el tiempo de cada tramo de la marcha ( $T$ ), se procede a calcular la velocidad que debe tener el motor para moverse desde una posición a otra en incrementos/segundo, utilizando las fórmulas de movimiento circular uniformemente variado [MCUV](#) (Ecuación [4.3](#)). Dónde  $\beta$  es el incremento necesario del motor,  $\dot{\beta}$  es la velocidad del motor y  $T$  es el tiempo de cada tramo.

$$\beta = \left( \frac{\dot{\beta} - \dot{\beta}_o}{2} \right) \cdot T, \quad \dot{\beta}_o = 0$$

$$\dot{\beta} = 2 \cdot \frac{\beta}{T} \left[ \frac{\text{inc}}{\text{s}} \right] \quad (4.3)$$

Para transformar de incrementos/segundo a **rpm** se multiplica por las fracciones respectivas:

$$\dot{\beta} \left[ \frac{\text{inc}}{\text{s}} \right] \cdot \frac{1 \text{ rev}}{4096 \text{ inc}} \cdot \frac{60 \text{ s}}{1 \text{ min}} = \dot{\beta} \cdot \frac{60}{4096} [\text{rpm}] \approx 0.145 \cdot \dot{\beta} [\text{rpm}]$$

El mismo cálculo se aplica para los motores de cada articulación del exoesqueleto. Cada tramo de la marcha tendrá una velocidad diferente acorde con su tiempo respectivo.

#### 4.4.2. Interpolación parabólica

Para generar las señales de referencia que van a seguir los motores de cada articulación, únicamente se ingresan los puntos máximos y mínimos de la referencia original que deben seguir los motores (Tablas 4.1, 4.2 y 4.3). Lo que se espera es que los motores lleguen a dichas posiciones de una manera suavizada, evitando movimientos bruscos que puedan afectar al usuario del exoesqueleto. Para lograr una trayectoria suavizada entre los puntos de referencia, se utiliza la interpolación parabólica que los controladores **EPOS4** tienen incorporada.

La interpolación parabólica une dos puntos consecutivos de la referencia de las articulaciones mediante dos funciones parabólicas y una función lineal. Esto se logra utilizando una aceleración y desaceleración constante que permite cambiar la velocidad del motor de cada articulación de manera uniforme. Con ello se obtiene una trayectoria suavizada que los motores siguen al momento de ejecutar un ciclo de caminata [14]. El comportamiento de la interpolación parabólica se muestra en la Figura 4.13.

Para lograr una mayor suavidad en la interpolación se necesita cumplir la Ecuación 4.4 en cada tramo de un ciclo de caminata para todas las articulaciones. Dónde  $\ddot{\beta}$  es la aceleración y desaceleración que debe tener el motor en cada tramo.

$$\ddot{\beta} = \frac{4 \cdot (\beta_f - \beta_o)}{T^2} = \frac{2 \cdot \dot{\beta}}{T} \left[ \frac{\text{rpm}}{\text{s}} \right] \quad (4.4)$$

Tanto la aceleración como la desaceleración de cada tramo tienen el mismo valor. Se utiliza el mismo valor para que las funciones parabólicas generadas entre los puntos de referencia de cada tramo tengan la misma forma tanto al inicio como al final de la curva, es decir, que la trayectoria no tenga una deformación.

Lo que se logra con esta interpolación es generar una curva suavizada entre dos puntos consecutivos de la referencia (cada tramo de la marcha). Por lo tanto, se aplica la interpolación a cada uno de los puntos de referencia ingresados. Es necesario tener una curva suavizada que

el motor de cada articulación debe seguir, para que el movimiento del exoesqueleto sea lo más natural y parecido a la caminata de una persona sin patología.

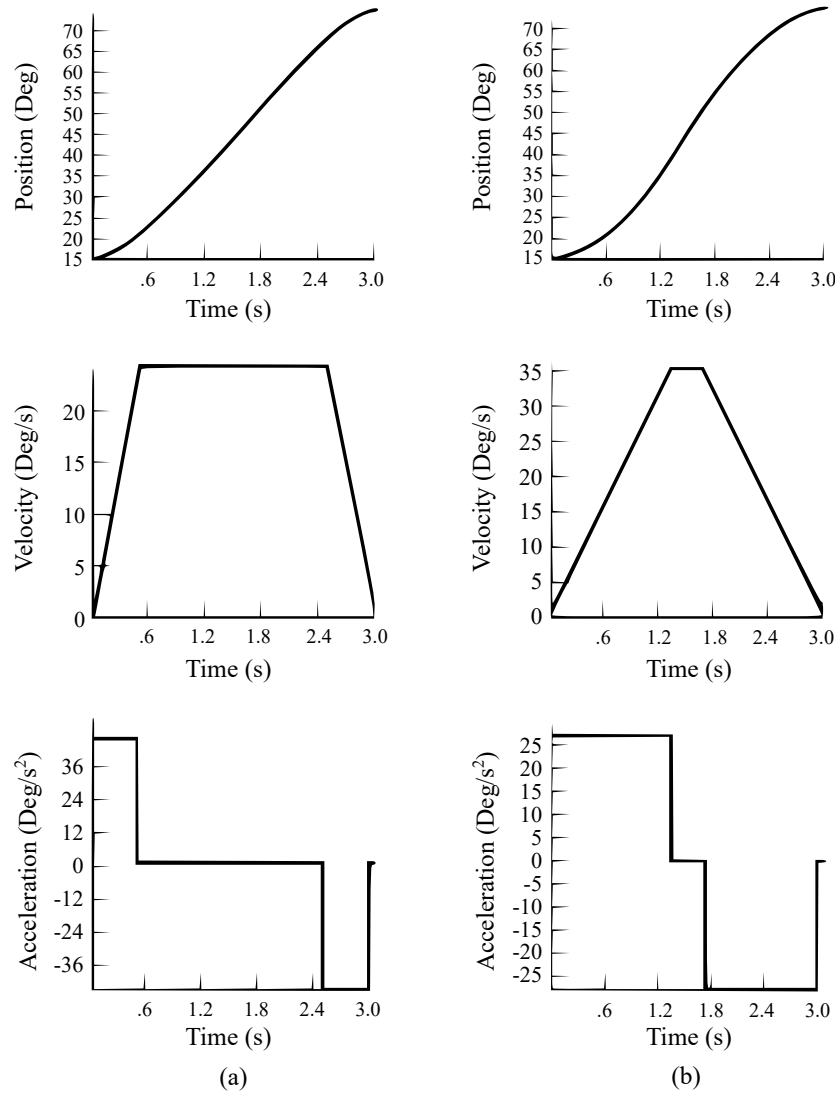


Figura 4.13: Perfiles de posición, velocidad y aceleración. (a) Gráficas con gran aceleración. (b) Gráficas con baja aceleración [14]

## 4.5. Programación para el control y comunicación del exoesqueleto de extremidad inferior

En esta sección se explica la programación necesaria para almacenar y enviar las señales de referencia que deben seguir los motores del exoesqueleto de extremidades inferiores dentro

de una Raspberry Pi 3, así como para recibir información de realimentación de los mismos. La Raspberry Pi 3 utiliza el sistema operativo *Raspbian*, y será el dispositivo maestro que se encargará de manejar los controladores mediante funciones propias del [EPOS4](#). Las funciones se obtienen por medio de una librería que los fabricantes *Maxon Motor* ponen a la disposición de manera gratuita. Esta librería es compatible con sistemas operativos como *Windows* y *Linux*. La librería es una implementación de protocolos que permiten la comunicación [CAN](#) entre los [EPOS4](#) y un computador (Raspberry Pi 3), y se los puede utilizar en diferentes plataformas de programación, una de ellas es C++.

Primero es necesario descargar e instalar la librería *EPOS\_Linux\_Library* dentro del dispositivo maestro, que es la Raspberry Pi 3. La instalación de esta librería se detalla en el Apéndice [D](#). La librería permitirá utilizar las funciones listadas en el documento *EPOS Command Library* [52]. Por medio de dichas funciones se puede enviar y recibir información desde y hacia los [EPOS4](#) colocados en cada articulación. Esto facilita el trabajo ya que únicamente será necesario ingresar ciertos parámetros en cada función, que a continuación se explicarán, para tener una comunicación directa con cada [EPOS4](#). Para mayor información acerca de las funciones que se utiliza en la elaboración de código que controla al exoesqueleto, se puede consultar el documento *EPOS Command Library*, en el cual se explica la utilidad y parámetros de cada una de las funciones.

Se necesita tener una plataforma de programación de software libre debido a que se va a trabajar en una Raspberry Pi 3 para el manejo de los controladores y los motores de cada articulación. Se ha escogido el lenguaje C++ por su fácil implementación y porque las funciones que permiten el uso de los controladores es compatible con este lenguaje. Para que la programación sea más amigable se utiliza un [IDE](#) como *CodeBlocks*.

Una vez calculados los parámetros que el exoesqueleto necesita para funcionar correctamente, se procede a ingresarlos en la programación que controla el exoesqueleto. En la Figura [4.14](#) se presenta un diagrama de bloques en el que se puede observar el algoritmo implementado en C++ para generar el movimiento, comunicación y control de los motores de cada articulación del exoesqueleto.

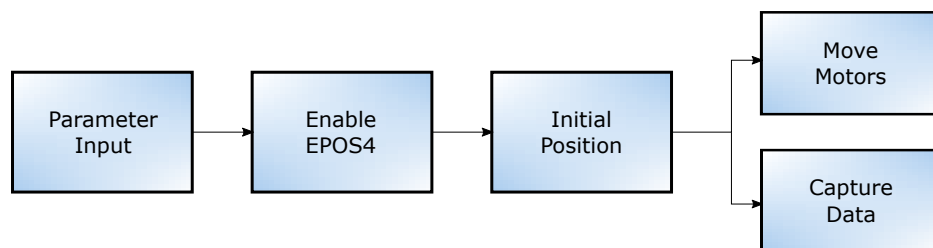


Figura 4.14: Programación implementada en el exoesqueleto

### 4.5.1. Ingreso de parámetros iniciales

Se debe aplicar las transformaciones descritas en la Sección 4.4.1, para ello se requiere que las trayectorias de referencia se encuentren dentro del código. Las trayectorias predefinidas describen el movimiento del miembro inferior izquierdo de una persona sana durante un ciclo de marcha, y guían el movimiento de los motores colocados en el exoesqueleto. Las trayectorias de referencia permiten que una persona que tiene algún problema en sus piernas, pueda imitar un patrón de marcha natural y se sienta cómoda realizando las terapias de rehabilitación.

Dentro del IDE de programación se ingresan los valores de referencia presentados en las Tablas 4.1, 4.2 y 4.3. Estos valores representan la posición angular (medida en grados) de cada uno de los tramos que se tiene durante un ciclo de caminata, con la duración de cada tramo. Los valores ingresados deben estar dentro de listas con el tipo de variable *long* para la posición y *double* para el tiempo, dichas variables serán compatibles con las funciones del EPOS4. Adicionalmente, se requiere una variable extra que almacena el valor de un ciclo de caminata necesario para los cálculos de velocidad y aceleración que necesitan los motores del exoesqueleto (Listado 4.1).

```
// Posicion de la cadera en grados
long TargetPosition [] = {30, -15, 35, 30};
// Posicion de la rodilla en grados
long TargetPosition2 [] = {-3, -13, -3, -65, -3};
// Posicion del tobillo en grados
long TargetPosition3 [] = {4, 18, -8, 24, 2, 8, 4};

// Porcentaje de caminata de la cadera
double CyclePercent [] = {0, 0.52, 0.87, 1};
// Porcentaje de caminata de la rodilla
double CyclePercent2 [] = {0, 0.18, 0.4, 0.69, 1};
// Porcentaje de caminata del tobillo
double CyclePercent3 [] = {0, 0.08, 0.38, 0.67, 0.79, 0.92, 1};

// Tiempo de ciclo de caminata
int HikingTime = 10;
```

Listado 4.1: Ingreso de parámetros iniciales

No es necesario ingresar toda la curva de referencia presentada en Sección 4.2, solo basta con tomar los puntos máximos y mínimos de la curva (Tablas 4.1, 4.2 y 4.3), ya que con las funciones del EPOS4 se logra realizar una interpolación parabólica (Sección 4.4.2) entre los puntos ingresados. La interpolación genera la curva de referencia deseada en las articulaciones del exoesqueleto.

Para generar el movimiento en el exoesqueleto se utiliza el modo de operación llamado *Profile Position Mode* que está implementado en cada uno de los EPOS4 de cada articulación. Gracias

a la librería instalada se puede utilizar ciertas funciones dentro de la programación de C++ que permiten utilizar este modo de operación. Lo que se logra con dichas funciones es abstraer el proceso de creación de las tramas que se envían por el bus CAN. Es decir, en las funciones se ingresan ciertos parámetros necesarios para el movimiento de los motores, y automáticamente se transforman los parámetros en tramas CAN listas para ser transmitidas a los EPOS4 por el bus de comunicación.

Desde las variables iniciales ingresadas en la programación que contienen las referencias de cada articulación, se procede a calcular el resto de parámetros que se requieren ingresar en las funciones del EPOS4, para que los motores giren de acuerdo a la referencia deseada. Primero se multiplica cada una de las referencias ingresadas por el valor calculado en las relaciones de la Sección 4.4.1 que corresponde a cada articulación. Así se logra transformar los valores de grados a incrementos (Listado 4.2).

```
// Calculo de incrementos que debe dar para llegar al angulo necesario
for (int y = 0; y <= 3; y++){
    // Incremento de la Cadera
    TargetPosition[y] = TargetPosition[y]*2230;
}
for (int y = 0; y <= 4; y++){
    // Incrementos de la Rodilla
    TargetPosition2[y] = TargetPosition2[y]*487;
}
for (int y = 0; y <= 6; y++){
    // Incrementos del tobillo
    TargetPosition3[y] = TargetPosition3[y]*487;
}
```

Listado 4.2: Transformaciones de unidades

Los motores de cada articulación deben generar movimientos suavizados, como las señales presentadas en la Sección 4.2, para que el movimiento general del exoesqueleto sea muy natural y parecido a una caminata humana. Para lograr el movimiento natural, se mencionó que los EPOS4 utilizan una interpolación parabólica, específicamente en el modo *Profile Position Mode*, que necesita de parámetros específicos para generar una trayectoria suavizada en los motores. Los parámetros necesarios en este modo de operación son la posición, velocidad y aceleración, y deben ser ingresados de manera adecuada para generar las trayectorias suavizadas. Se deben calcular velocidades y aceleraciones independientes para cada uno de los tramos de las trayectorias de cada articulación. Primero es necesario calcular el tiempo que tarda cada uno de los tramos de las trayectorias, y con estos tiempos calcular la velocidad y aceleración que los motores deben utilizar en dichos tramos. Se utiliza un algoritmo que recorre las listas de los porcentajes de caminata de cada articulación, se multiplica por el tiempo de caminata total y se obtiene el tiempo de cada tramo de la trayectoria. Posteriormente, a los tiempos de cada tramo se les aplica las ecuaciones presentadas en las Secciones 4.4.1 y 4.4.2. Todo el



algoritmo es ingresado en la programación de C++ para cada una de las articulaciones. Los valores de velocidad y aceleración deben ser almacenados en listas con una variable del tipo *unsigned int*, la cual es compatible con las funciones del EPOS4. En el Listado 4.3 se presenta la implementación en C++ para la cadera. Es el mismo procedimiento para la rodilla y tobillo, únicamente hay que tomar en cuenta que el número de tramos varía en cada articulación.

```
// Calculo de velocidad y aceleracion de cada tramo - Cadera
for (int i = 1; i <= 3; i++){
    // Tiempo de cada tramo de la trayectoria
    Tf = HikingTime * (CyclePercent[i] - CyclePercent[i - 1]);
    // Velocidad de cada tramo de la trayectoria.
    Vf = 2*(TargetPosition[i] - TargetPosition[i - 1]) / (Tf);
    // Guardar la velocidad de cada tramo
    ProfileVelocity[i-1] = rint(abs(Vf*0.0145));
    // Guardar la aceleracion y desaceleracion de cada tramo
    ProfileAcceleration[i-1] = rint(2*ProfileVelocity[i-1]/Tf);
    ProfileDeceleration[i-1] = ProfileAcceleration[i-1];
}
```

Listado 4.3: Cálculo de velocidades y aceleraciones para la cadera

Todo el cálculo de los parámetros que los motores necesitan para generar el movimiento se realiza antes de iniciar el movimiento de la pierna. Mientras se esté ejecutando el programa, los parámetros se almacenarán en la memoria RAM de la Raspberry Pi 3. Cuando el exoesqueleto necesite realizar el movimiento de las articulaciones, se debe leer los valores de la memoria RAM y enviarlos a los EPOS4 correspondientes para generar el movimiento.

#### 4.5.2. Habilitar el EPOS4

En este punto se crea una primera comunicación entre la Raspberry Pi 3 y los EPOS4 de cada una de las articulaciones. Primero se necesita definir el nombre de los equipos, el protocolo, la interfaz y el puerto que se va a utilizar para la comunicación. Además se necesita crear variables que almacenen el valor del identificador de los tres EPOS4 (Listado 4.4).

```
// Parametros para comunicacion principal
string g_deviceName = "EPOS4";
string g_protocolStackName = "CANopen";
string g_interfaceName = "CAN_mcp251x 0";
string g_portName = "CAN0";
void* g_pKeyHandle;
unsigned int p_pErrorCode;

unsigned short g_usNodeId = 1; // ID (Hip)
unsigned short g_usNodeId2 = 2; // ID (Knee)
unsigned short g_usNodeId3 = 3; // ID (Ankle)
```

Listado 4.4: Comunicación principal

De las variables creadas, se extraen únicamente los caracteres que son los parámetros que necesita la función llamada *OpenDevice*. Esta función permite abrir un puerto de comunicación entre la Raspberry Pi 3 y los tres EPOS4. Una vez creada la comunicación entre todos los dispositivos de la red CAN, se utiliza una nueva función llamada *SetEnableState* que permite activar los dispositivos EPOS4 (una por cada articulación). De esta manera los controladores están listos para recibir y enviar mensajes según el modo de operación que se desee utilizar (Listado 4.5).

```
// Extraer caracteres
char* pDeviceName = new char[255];
char* pProtocolStackName = new char[255];
char* pInterfaceName = new char[255];
char* pPortName = new char[255];

strcpy(pDeviceName, g_deviceName.c_str());
strcpy(pProtocolStackName, g_protocolStackName.c_str());
strcpy(pInterfaceName, g_interfaceName.c_str());
strcpy(pPortName, g_portName.c_str());
// Abrir el puerto para comunicarse con los EPOS4
g_pKeyHandle = VCS_OpenDevice(pDeviceName, pProtocolStackName, pInterfaceName,
    pPortName, &p_pErrorCode);
// Habilitar EPOS4 de la cadera
VCS_SetEnableState(g_pKeyHandle, g_usNodeId, &p_pErrorCode);
// Habilitar EPOS4 de la rodilla
VCS_SetEnableState(g_pKeyHandle, g_usNodeId2, &p_pErrorCode);
// Habilitar EPOS4 del tobillo
// VCS_SetEnableState(g_pKeyHandle, g_usNodeId3, &p_pErrorCode);
```

Listado 4.5: Extracción de caracteres

### 4.5.3. Posición inicial

Antes de iniciar el movimiento del exoesqueleto para un ciclo de caminata, es necesario colocarlo en una posición inicial. La posición inicial es el primer valor que se encuentra en las referencias ingresadas de cada articulación, es decir, el tobillo empieza en 4 grados, la rodilla en 3 grados y la cadera en 30 grados en el plano sagital. En este punto se utiliza el modo *Profile Position Mode*, para ello lo primero es activar este modo de operación en los EPOS4 de cada articulación con la función *ActivateProfilePositionMode*. Luego se cargan los parámetros que el modo de operación necesita, como la velocidad, aceleración y desaceleración que tendrán los motores mediante la función *SetPositionProfile*. Los parámetros para *SetPositionProfile* toman el valor de 500 tanto para velocidad como aceleración. En este caso no importan los valores de los parámetros porque es la posición inicial que debe tomar el exoesqueleto. Finalmente, se envía a los motores la posición inicial de sus referencias mediante la función *MoveToPosition*

para que los mismos giren a dicha posición. Las funciones se aplican a los motores de las tres articulaciones, para llevar todos a su posición inicial (Listado 4.6).

```
// Posicion inicial de la cadera
// Activar el modo de ProfilePositionMode en el EPOS4 de la cadera
VCS_ActivateProfilePositionMode(g_pKeyHandle, g_usNodeId, &p_pErrorCode);
// Ingresar parametros para el movimiento inicial de la cadera
VCS_SetPositionProfile(g_pKeyHandle, g_usNodeId, 500, 500, 500, &p_pErrorCode);
// Mover la cadera a la posicion inicial de su referencia
VCS_MoveToPosition(g_pKeyHandle, g_usNodeId, TargetPosition[0], 1, 1, &
    p_pErrorCode);
```

Listado 4.6: Posición inicial

#### 4.5.4. Multiprocesos

En esta parte del código empieza el movimiento del exoesqueleto de acuerdo a los parámetros ingresados en los pasos anteriores. Lo que se hace es utilizar cuatro hilos o multiprocesos para ejecutar cuatro códigos a la vez. Tres hilos se utilizan para ejecutar el movimiento de las tres articulaciones del exoesqueleto y el último hilo se utiliza para capturar la información relevante de cada una de las articulaciones del exoesqueleto.

En los tres hilos que envían la información para mover los motores de cada articulación se requiere nuevamente utilizar las funciones del modo *Profile Position Mode* que se utilizó para mover el exoesqueleto a su posición inicial. Como el modo de operación ya quedó activado en la sección de código que coloca el exoesqueleto en su posición inicial, en los hilos únicamente se utiliza las funciones *SetPositionProfile* y *MoveToPosition*. Lo que realiza cada hilo es, mediante un bucle, recorrer las listas que contienen los valores de las velocidades y aceleraciones calculados anteriormente para cada tramo de cada articulación. Se agregan los parámetros tanto de velocidad como de aceleración en la función *SetPositionProfile* en cada iteración del bucle. Posteriormente, se envía el valor de la referencia que corresponde a dichos parámetros de velocidad y aceleración mediante *MoveToPosition* y se genera una pausa con el tiempo que demora en recorrer el motor de la posición anterior a la nueva. Se realiza el mismo proceso para todos los valores de referencia de cada una de las articulaciones. El EPOS4 automáticamente generará la interpolación parabólica con los parámetros de velocidad y aceleración calculados en cada tramo para que el motor gire formando la trayectoria suavizada (Listado 4.7).

```
// Hilo para el movimiento de la cadera
for (int x = 1; x <= 3; x++){
    // Cargar parametros de cada tramo
    VCS_SetPositionProfile(g_pKeyHandle, g_usNodeId, ProfileVelocity[x-1],
        ProfileAcceleration[x-1], ProfileDeceleration[x-1], &p_pErrorCode);
    // Mover la cadera a la posicion deseada
    VCS_MoveToPosition(g_pKeyHandle, g_usNodeId, TargetPosition[x], 1, 1, &
        p_pErrorCode);
```

```
// Esperar hasta que la cadera termine el tramo
usleep((HikingTime * (CyclePercent[x] - CyclePercent[x - 1]))*1000000);
}
```

Listado 4.7: Hilo para movimiento de cadera

Para el desarrollo de este proyecto, es necesario conocer algunos valores relevantes que va a tener cada articulación, como: la posición de las articulaciones, la velocidad con la que están girando las articulaciones, la corriente y torque que generan los motores de cada una de ellas. El último hilo será el encargado de capturar los datos relevantes que gracias a los sensores *hall* y *encoder* de cada motor, y las funciones del [EPOS4](#) se pueden obtener de una manera sencilla.

Hay que tener presente que al momento de recibir los valores de posición que entrega el *encoder* de los motores, estos entregan en unidades de incrementos. Es necesario utilizar la relación inversa presentada en la Sección 4.4.1 para obtener el dato en grados y sea de fácil entendimiento para el operador del exoesqueleto.

Para indicar el torque de cada uno de los motores, cada uno tiene en sus características técnicas una constante de torque/corriente. Esta constante será multiplicada por el dato de la corriente capturada durante el movimiento del exoesqueleto.

Los [EPOS4](#) capturan la información relevante del exoesqueleto cada instante de tiempo. Existen funciones que permiten extraer esta información de manera sencilla.

- `GetObject` → Entrega el valor de referencia del motor
- `GetPositionIs` → Entrega la posición actual del motor
- `GetVelocityIs` → Entrega la velocidad actual del motor
- `GetCurrentIs` → Entrega la corriente actual consumida del motor

Dentro del hilo final se ingresa las funciones del [EPOS4](#) para la captura de datos de cada uno de los motores. Cada motor utiliza las cuatro funciones listadas anteriormente para obtener el valor de referencia que sigue, el movimiento generado, la velocidad producida y corriente generada en cada uno de los motores. En total se ingresan 12 funciones dentro del hilo para los tres motores y se ejecutan una a continuación de otra (Listado 4.8).

```
// Obtener la posicion de la cadera
VCS_GetPositionIs(g_pKeyHandle, g_usNodeId, &pPositionIs, &p_pErrorCode);
// Obtener la referencia de la cadera
VCS_GetObject(g_pKeyHandle, g_usNodeId, 0x6062, 0x00, &pDemandIs, 4, &pNbBytes,
&p_pErrorCode);
// Obtener corriente de la cadera
VCS_GetCurrentIs(g_pKeyHandle, g_usNodeId, &pCurrentIs, &p_pErrorCode);
// Obtener velocidad de la cadera
VCS_GetVelocityIs(g_pKeyHandle, g_usNodeId, &pVelocityIs, &p_pErrorCode);
```

Listado 4.8: Extraer datos de la cadera

## Capítulo 5

# Pruebas y resultados

La versión final del exoesqueleto de extremidades inferiores se puede ver en la Figura 5.1a. Se puede apreciar el detalle y ubicación de los motores y placas de control en sus respectivas articulaciones. Además, se han colocado carcasas plásticas en cada una de las articulaciones para ocultar todo el cableado, y que el exoesqueleto tenga una mejor apariencia. En este punto el exoesqueleto está listo para realizar sus pruebas de funcionamiento y, evaluar así, el comportamiento del control y la red de comunicaciones implementada.

Para realizar las pruebas del exoesqueleto se requiere que el mismo esté colocado en la posición inicial o posición cero como se indica en la Figura 5.1b. El exoesqueleto debe conectarse al dispositivo central, que es la Raspberry Pi 3, mediante un cable RJ-9 para la comunicación, y también debe ser conectado a la fuente de alimentación para energizar todo el circuito. Una vez conectado todo el sistema, se procede a ejecutar el código implementado en la Raspberry Pi 3 para accionar el movimiento del exoesqueleto.

Se realizaron diferentes pruebas sobre el exoesqueleto, en primer lugar variando el tiempo de ciclo de caminata para el movimiento del exoesqueleto en diferentes escenarios. Esto permite analizar el comportamiento del control PID implementado en cada una de las articulaciones al seguir una referencia correspondiente. Posteriormente se realizaron pruebas que permitieron observar el desempeño de la red CAN tanto para enviar información a cada una de las articulaciones como para recibir información de realimentación de las mismas.



(a) Ensamblado final del exoesqueleto

(b) Posición inicial del exoesqueleto

Figura 5.1: Exoesqueleto de extremidades inferiores

Los resultados se dividen en una sección para indicar las gráficas de control obtenidas en cada una de las pruebas aplicadas al exoesqueleto. Y una segunda sección para indicar el funcionamiento y rendimiento de la red comunicaciones.

## 5.1. Evaluación de los algoritmos de control

Se realizan tres pruebas para contrastar el correcto funcionamiento del control [PID](#). Para las primeras pruebas se ingresan constantes del control [PID](#) de manera manual. Luego se utiliza el *software EPOS Studio* para realizar el ajuste de las constantes de control de manera automática con la herramienta *autotuning*. Finalmente, se realizan pruebas agregando un peso de 15 Kg a la estructura del exoesqueleto para analizar el comportamiento del control [PID](#) con carga. En cada una de estas pruebas se varía el tiempo del ciclo de la marcha que sigue el exoesqueleto. Los tiempos varían entre 10, 7 y 4 segundos, lo que a su vez produce que la velocidad y aceleración de los motores en cada articulación del exoesqueleto sean variables. Con pequeños valores de tiempos en el ciclo de marcha, la velocidad y aceleración de los motores incrementará, mientras que al aumentar el tiempo de marcha los motores disminuirán en velocidad y aceleración. La

variación de los tiempos permite evaluar la rapidez y el comportamiento del control [PID](#) en diferentes escenarios. Al ejecutar el programa que permite realizar el movimiento y captura de datos sobre la posición de cada articulación del exoesqueleto, se almacena la información de cada articulación en un archivo CSV. Este archivo se exporta posteriormente a MATLAB para obtener las gráficas de las trayectorias generadas por las articulaciones. Se han realizado 10 pruebas con cada escenario obteniendo resultados similares en cada prueba, así que se ha seleccionado únicamente una prueba de cada escenario para presentarlos a continuación.

### 5.1.1. Control con constantes PID ingresadas manualmente

Para evaluar la eficiencia y/o utilidad del ajuste automático que *EPOS Studio* ofrece para el cálculo de las constantes [PID](#), primero se realizan pruebas ingresando constantes de manera manual en cada uno de los [EPOS4](#). Las constantes implementadas en cada articulación se presentan a continuación, las cuales han sido escogidas de manera experimental tratando de generar oscilaciones en el movimiento de las articulaciones sin llegar a la inestabilidad.

Cadera:

- $P = 2000 \text{ mA/rad}$ .
- $I = 20000 \text{ mA/rad}\cdot\text{s}$ .
- $D = 100 \text{ mA}\cdot\text{s/rad}$

Rodilla:

- $P = 5000 \text{ mA/rad}$ .
- $I = 20000 \text{ mA/rad}\cdot\text{s}$ .
- $D = 50 \text{ mA}\cdot\text{s/rad}$

Tobillo:

- $P = 150 \text{ mA/rad}$ .
- $I = 60000 \text{ mA/rad}\cdot\text{s}$ .
- $D = 50 \text{ mA}\cdot\text{s/rad}$

Se realizan tres pruebas y se obtiene la gráfica de cada una de las articulaciones para tiempos de 10, 7 y 4 segundos en el ciclo de marcha respectivamente.

En las Figuras [5.2a](#), [5.2b](#) y [5.2c](#) se presentan los resultados obtenidos de las articulaciones de la cadera, rodilla y tobillo, respectivamente, al mover el exoesqueleto con un ciclo de caminata de 10 s. El control no está ajustado y esto se evidencia claramente en las articulaciones de la rodilla y tobillo que oscilan alrededor de la señal de referencia. Aunque el exoesqueleto sigue la referencia, no lo hace de la mejor manera ya que las oscilaciones podrían ocasionar problemas futuros en pacientes que estén utilizando el exoesqueleto. La cadera es la única articulación que

no genera errores significativos a pesar de que también tiene un ajuste manual de sus constantes de control de control.

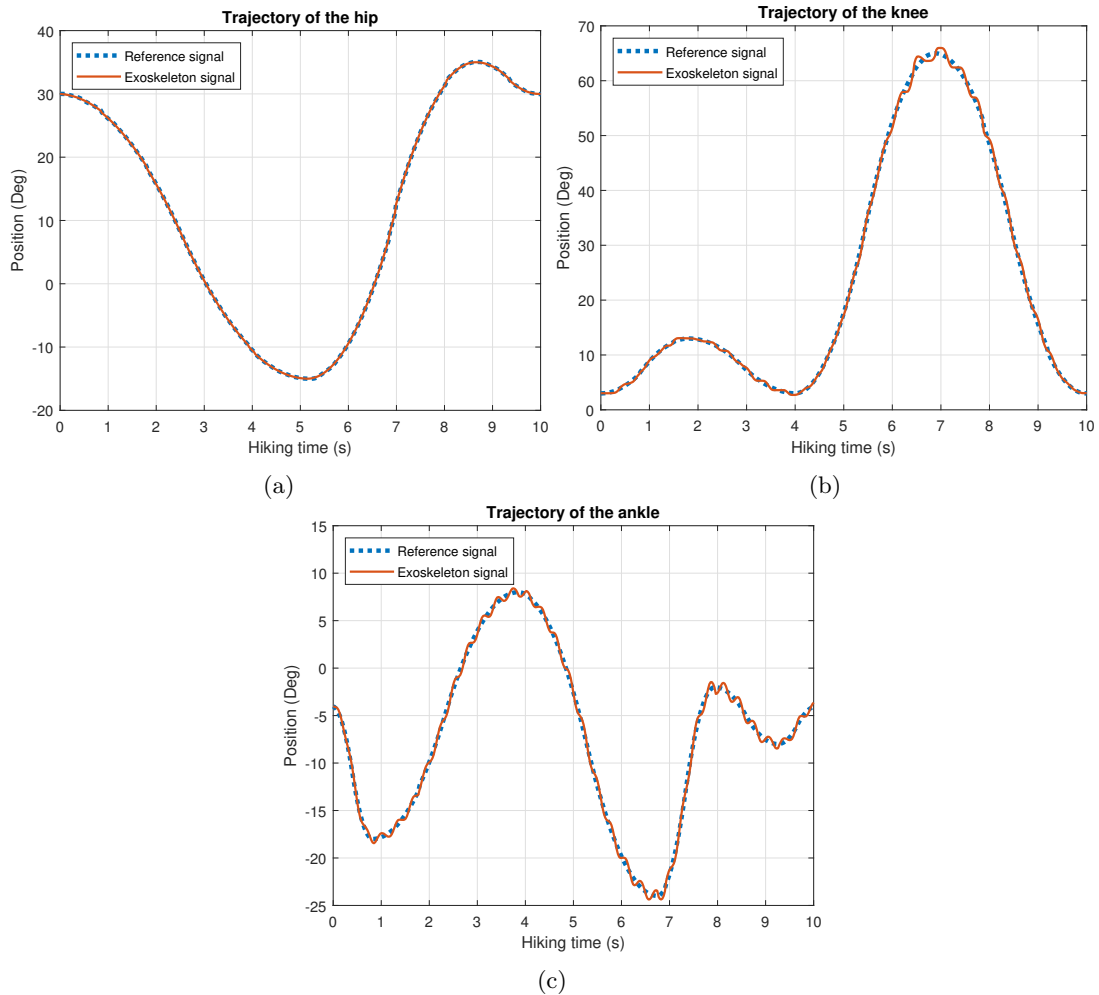


Figura 5.2: Trayectorias de caminata con 10 s y con un ajuste manual de constantes de control

A continuación, se procede a realizar pruebas con un tiempo de ciclo de marcha de 7 s. En las Figuras 5.3a, 5.3b y 5.3c se presenta la trayectoria generada por la articulación de la cadera, rodilla y tobillo respectivamente. El comportamiento es similar al movimiento con 10 segundos. Se producen oscilaciones en las articulaciones que podrían causarle daño e incomodidad al paciente que utilice el exoesqueleto. Una vez más, la cadera tiene un comportamiento aceptable a pesar de tener un ajuste manual en sus constantes de control. Además, la cadera no presenta errores significativos en su trayectoria a pesar que su movimiento es más veloz con respecto al tiempo de 10 s.

Finalmente se realizan las pruebas con un ciclo de caminata de 4 s. En las Figuras 5.4a, 5.4b



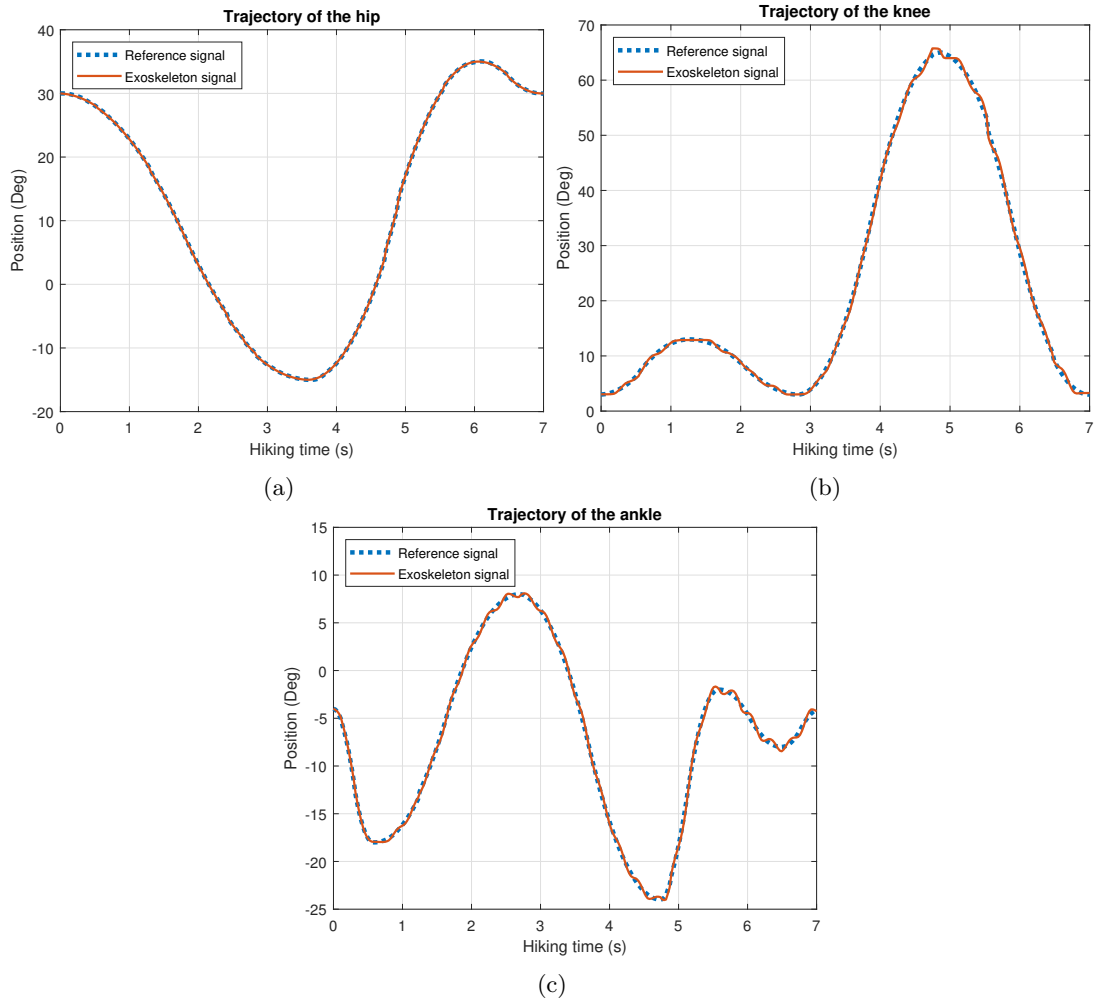


Figura 5.3: Trayectorias de caminata con 7 s y con un ajuste manual de constantes de control

y 5.4c se presentan los resultados obtenidos. Se puede ver que el ajuste manual de las constantes del control PID trabajan de mejor manera a movimientos rápidos que genera el exoesqueleto sobre la articulación de la cadera y del tobillo. No existe una oscilación excesiva alrededor de la referencia al realizar un ciclo de marcha con un tiempo de 4 s. Los resultados no son perfectos, pero las trayectorias generadas no presentan movimientos bruscos en las articulaciones de la cadera y tobillo. No se puede decir lo mismo de la rodilla, ya que al final del ciclo de marcha la articulación oscila sobre su referencia causando inestabilidad.

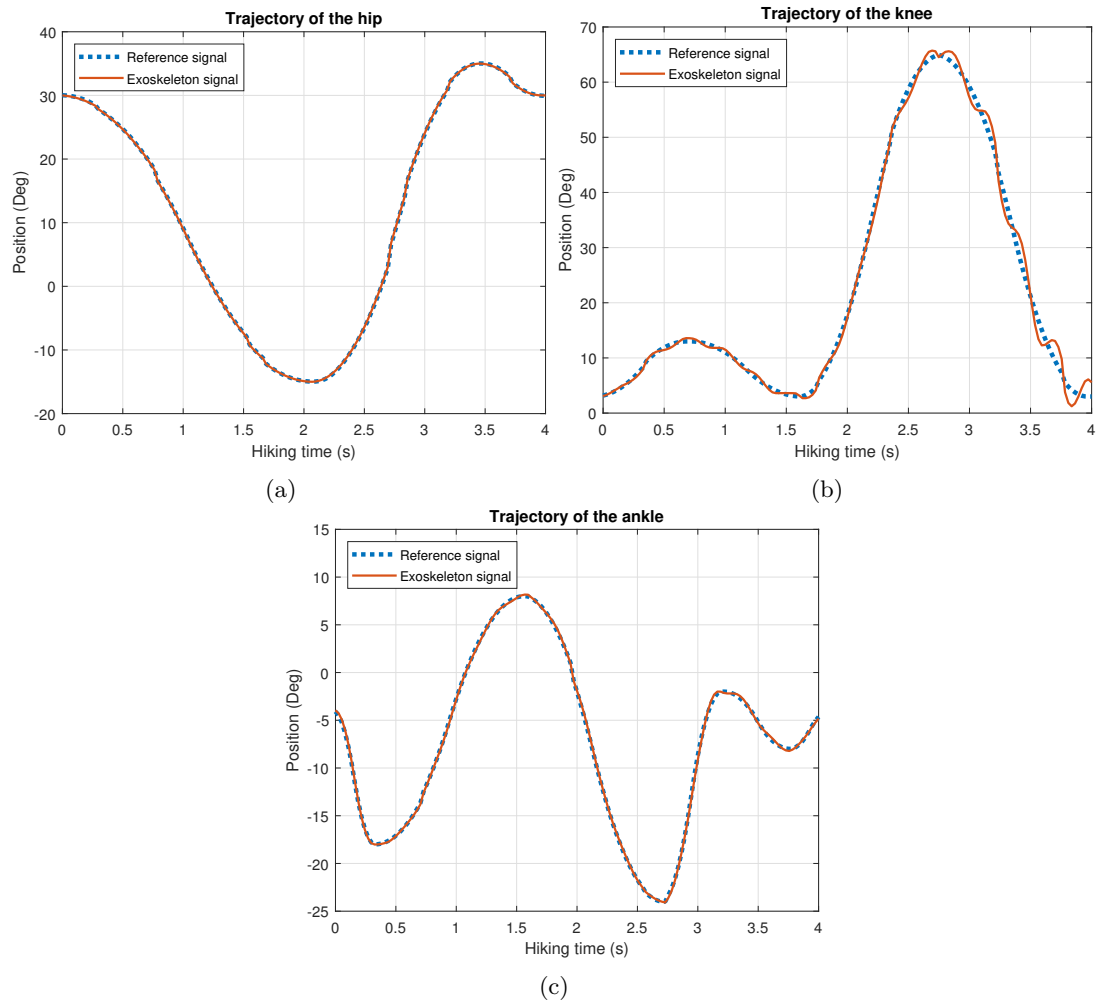


Figura 5.4: Trayectorias de caminata con 4 s con un ajuste manual de constantes de control

### 5.1.2. Control con ajuste automático de las constantes PID

Para este caso, las constantes para el controlador PID de cada una de las articulaciones son calculadas automáticamente por la herramienta *autotuning* de *EPOS Studio*, de acuerdo al siguiente detalle:

Cadera:

- $P = 13145.651 \text{ mA/rad}$ .
- $I = 33408.562 \text{ mA/rad}\cdot\text{s}$ .
- $D = 431.048 \text{ mA}\cdot\text{s/rad}$

Rodilla:

- $P = 7598.309 \text{ mA/rad}$ .
- $I = 37163.492 \text{ mA/rad}\cdot\text{s}$ .
- $D = 129.460 \text{ mA}\cdot\text{s/rad}$

Tobillo:

- $P = 12815.109 \text{ mA/rad}$ .
- $I = 99264.220 \text{ mA/rad}\cdot\text{s}$ .
- $D = 112.683 \text{ mA}\cdot\text{s/rad}$

Igual que en la sección anterior, se realizan tres pruebas diferentes con tiempos de 10, 7 y 4 s en los ciclos de marcha.

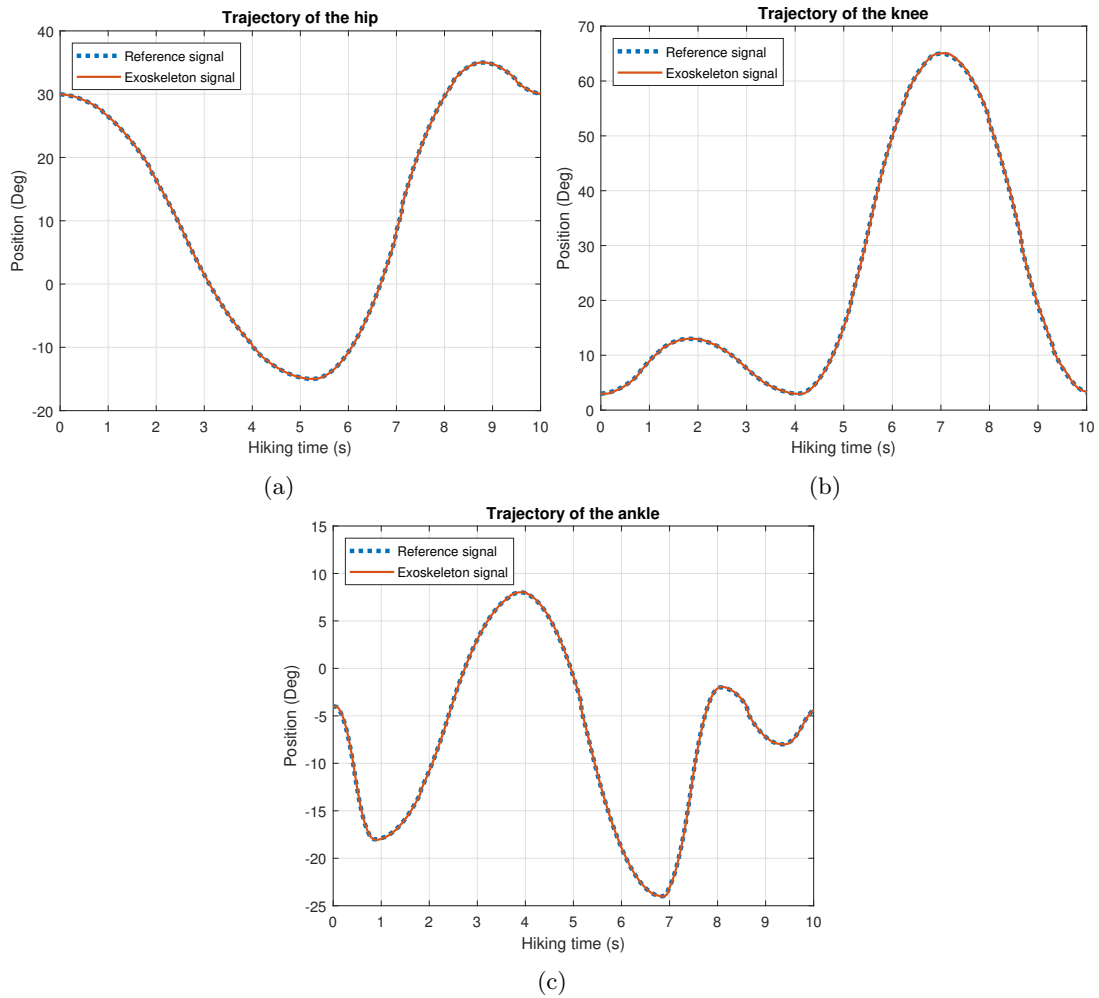


Figura 5.5: Trayectorias de caminata con 10 s y con un ajuste automático de las constantes de control

Primero se ejecuta la trayectoria de caminata para cada una de las articulaciones del exoes-

queleto con un tiempo de 10 s. En las Figuras 5.5a, 5.5b y 5.5c se puede ver la señal de referencia y los movimientos que el exoesqueleto produce durante un ciclo de marcha en cada una de sus articulaciones. Con el tiempo de 10 s se observa que el control PID aplicado a cada articulación funciona de manera ideal, es decir, la señal que sigue el exoesqueleto es casi idéntica a su referencia. Se puede decir que para ciclos de marcha relativamente lentos, el controlador implementado en los EPOS4 funciona sin errores cuando no se tiene peso en el exoesqueleto.

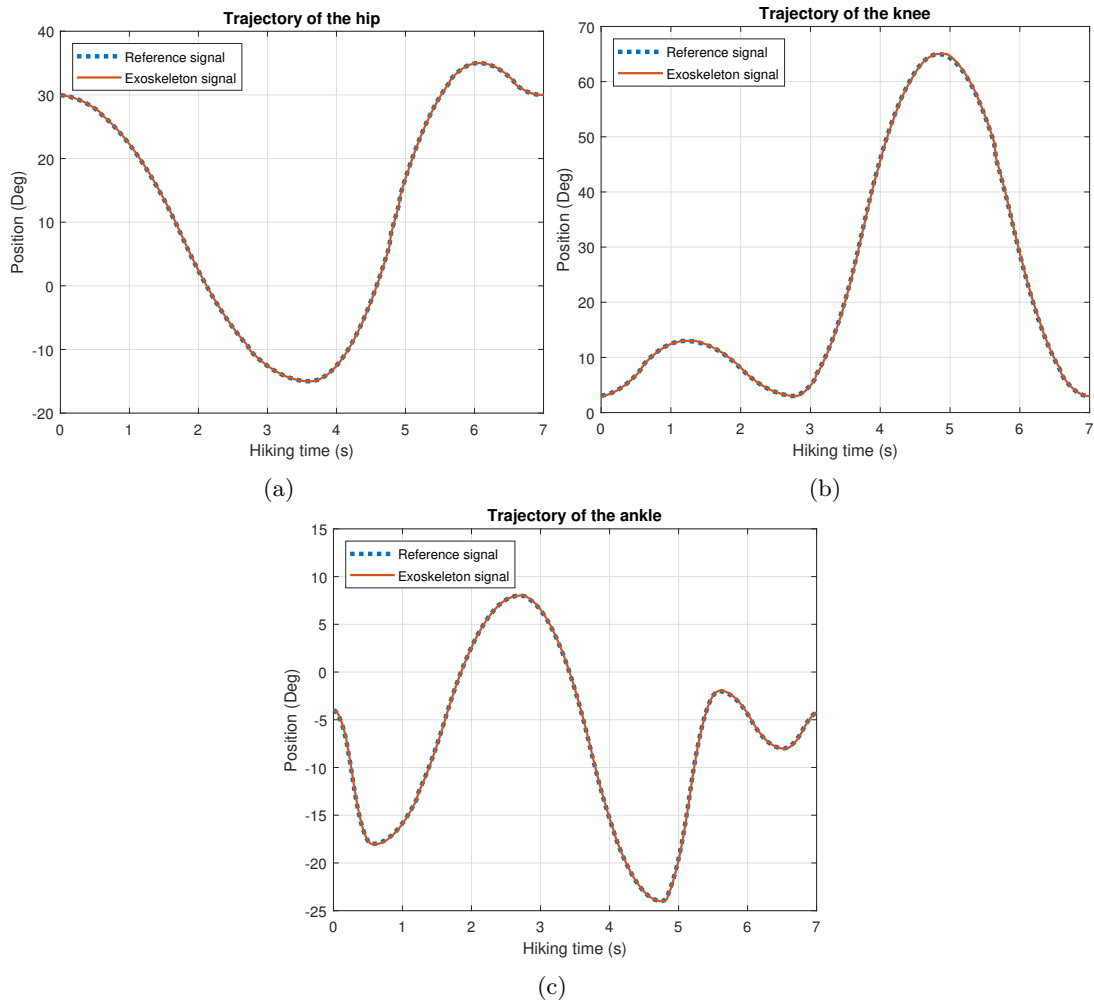


Figura 5.6: Trayectorias de caminata con 7 s con un ajuste automático de las constantes de control

El resultado para el segundo conjunto de pruebas se puede ver en las Figuras 5.6a, 5.6b y 5.6c que corresponde a un ciclo de caminata de 7 s. Con este tiempo se observa nuevamente el correcto funcionamiento del control implementado en el EPOS4 de cada articulación, ya que el error es prácticamente cero. Igual que el movimiento con 10 s, el control PID ayuda a que cada una de las articulaciones siga de manera idéntica a sus referencias correspondientes en todas

las articulaciones del exoesqueleto.

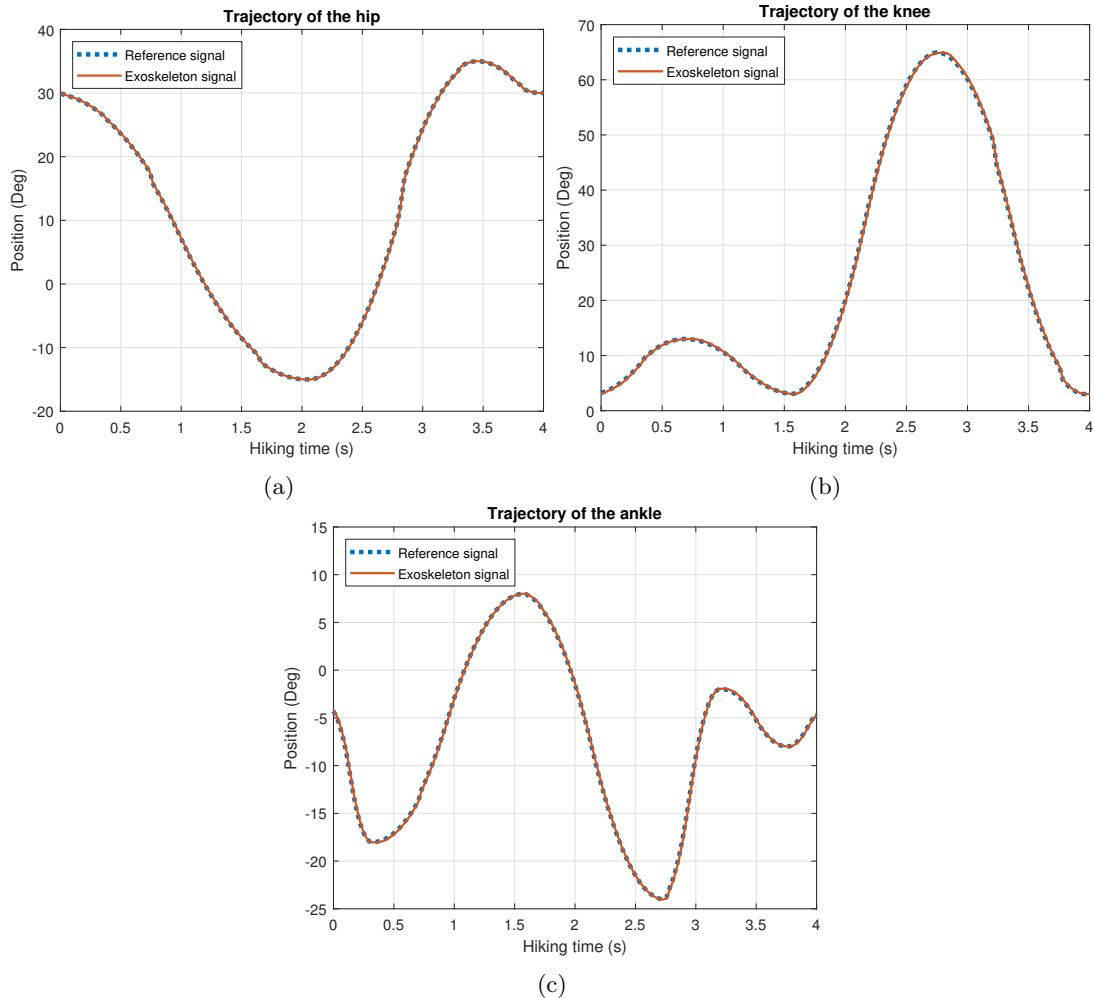


Figura 5.7: Trayectorias de caminata con 4 s con un ajuste automático de las constantes de control

Finalmente, en las Figuras 5.7a, 5.7b y 5.7c se presentan las trayectorias generadas por las articulaciones de la cadera, rodilla y tobillo respectivamente para un ciclo de caminata de 4 s. En este caso el tiempo de caminata es menor en comparación con las pruebas anteriores, y se puede ver que el control funciona pero genera errores ligeramente mayores que las pruebas anteriores. Existe un incremento de  $0.1^\circ$  en el error de cada articulación, que no afectan de manera significativa al funcionamiento del exoesqueleto. Se puede decir que el controlador implementado en los EPOS4 pierde ligeramente su precisión para los movimientos con tiempos de caminata relativamente pequeños.

### 5.1.3. Control con carga

Finalmente se realizan pruebas del exoesqueleto con carga. Se coloca un peso total de 15 kg sobre el exoesqueleto distribuidos con 5 kg en cada articulación. Esto con el fin de observar el comportamiento del control al tratar de mover una carga de masa semejante a la de una extremidad inferior. Se utilizan las mismas constantes de la sección anterior, que fueron calculadas automáticamente con la herramienta *autotuning* de *EPOS Studio*. Igual que con las pruebas de las secciones anteriores, se realiza la evaluación con ciclos de marcha de 10, 7 y 4 s.

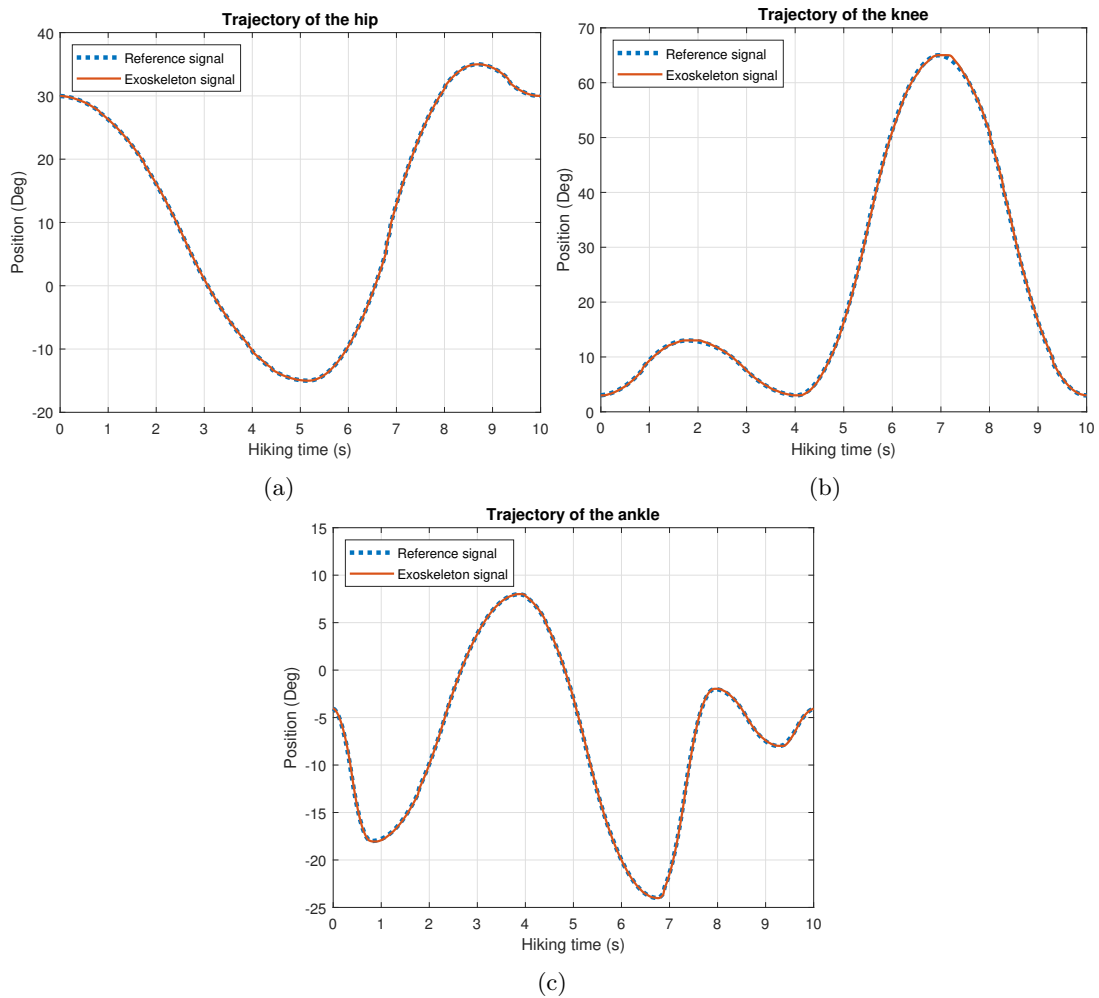


Figura 5.8: Trayectorias de caminata con 10 s y con una carga total de 15 kg

En las Figuras 5.8a, 5.8b y 5.8c se pueden ver las trayectorias generadas por las articulaciones de cadera, rodilla y tobillo respectivamente con un ciclo de caminata de 10 s y utilizando una carga total de 15 kg. Se verifica la eficiencia del controlador PID, ya que el exoesqueleto

presenta un comportamiento similar que el caso sin carga. Las trayectorias generadas por las articulaciones son casi idénticas a sus referencias con pequeñas excepciones. Es notable, además, un pequeño error aproximadamente a los 7.2 s de la trayectoria de la rodilla (Figura 5.8b), pero el control PID actúa de forma inmediata para corregir el mismo. El sistema de control es eficiente ya que a pesar del peso que se encuentra sobre el exoesqueleto, éste trata de corregir los errores rápidamente para que sean imperceptibles.

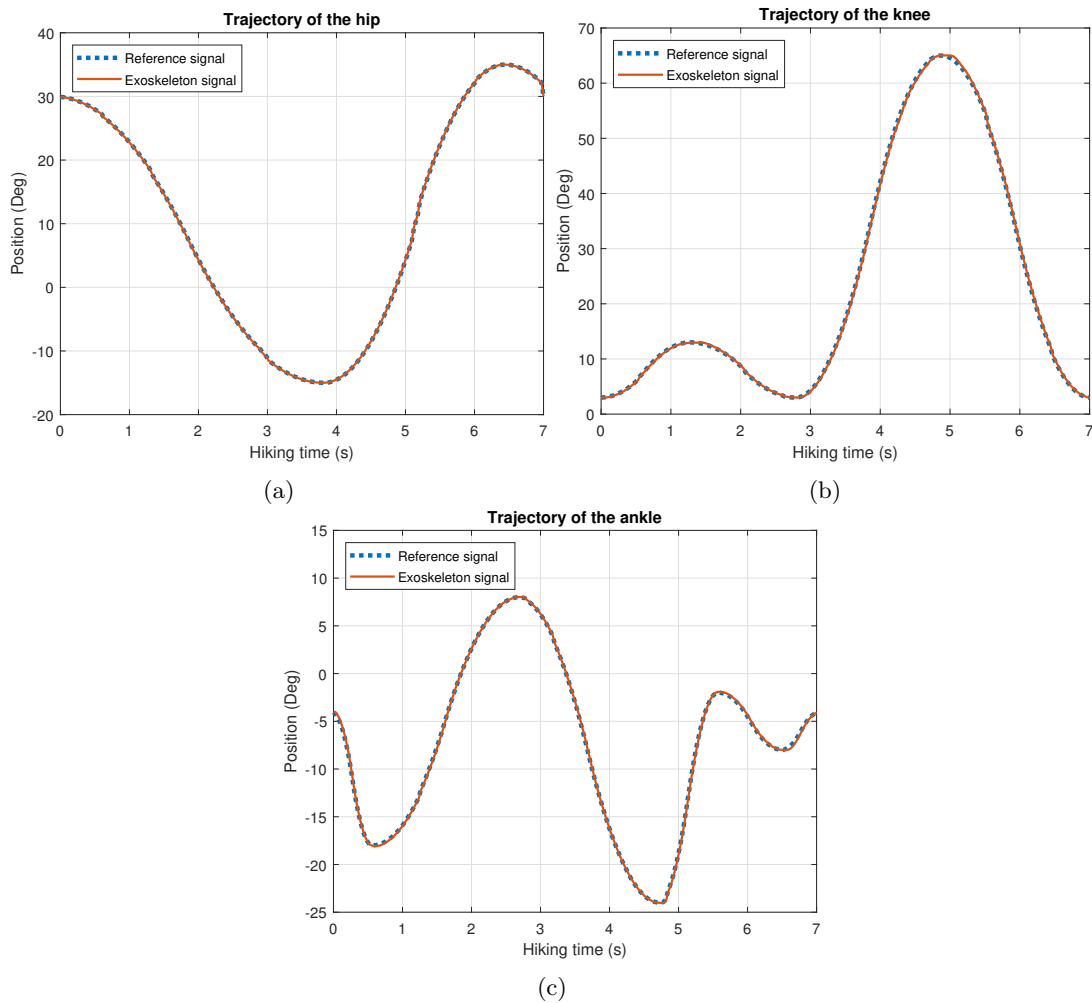


Figura 5.9: Trayectorias de caminata con 7 s y una carga total de 15 kg

Se obtiene un resultado similar al realizar el movimiento del exoesqueleto con un tiempo de caminata de 7 s. Las trayectorias generadas (Figuras 5.9a, 5.9b y 5.9c) indican que las articulaciones de la cadera, rodilla y tobillo siguen de manera casi idéntica a sus respectivas referencias. Únicamente en el tobillo (Figura 5.9c) se observa un ligero error en los tiempos de 2.8 y 5.7 s aproximadamente, pero el control PID es rápido y corrige el mismo para que la

articulación se ajuste a su referencia. Se observa que el error no es considerable, por lo que no afectaría a un paciente si estuviera utilizando el exoesqueleto. En general se puede concluir que los errores presentados pueden pasar desapercibidos debido a su bajo impacto en el seguimiento de la trayectoria.

Las evaluaciones finales se realizan con un ciclo de caminata de 4 s. Una vez más se puede ver en las Figuras 5.10a, 5.10b y 5.10c que el control PID implementado funciona correctamente en cada una de las articulaciones. Como en los casos anteriores, las trayectorias de cada articulación siguen su respectiva referencia de manera ideal. Se puede ver claramente que el exoesqueleto se mueve de una manera natural y muy suavizada como es requerido, a pesar de la carga que está sobre el mismo. Pero también se observan pequeños errores, que aunque no afecten de manera significativa al movimiento del exoesqueleto, son más notables que las trayectorias presentadas con 10 y 7 s.

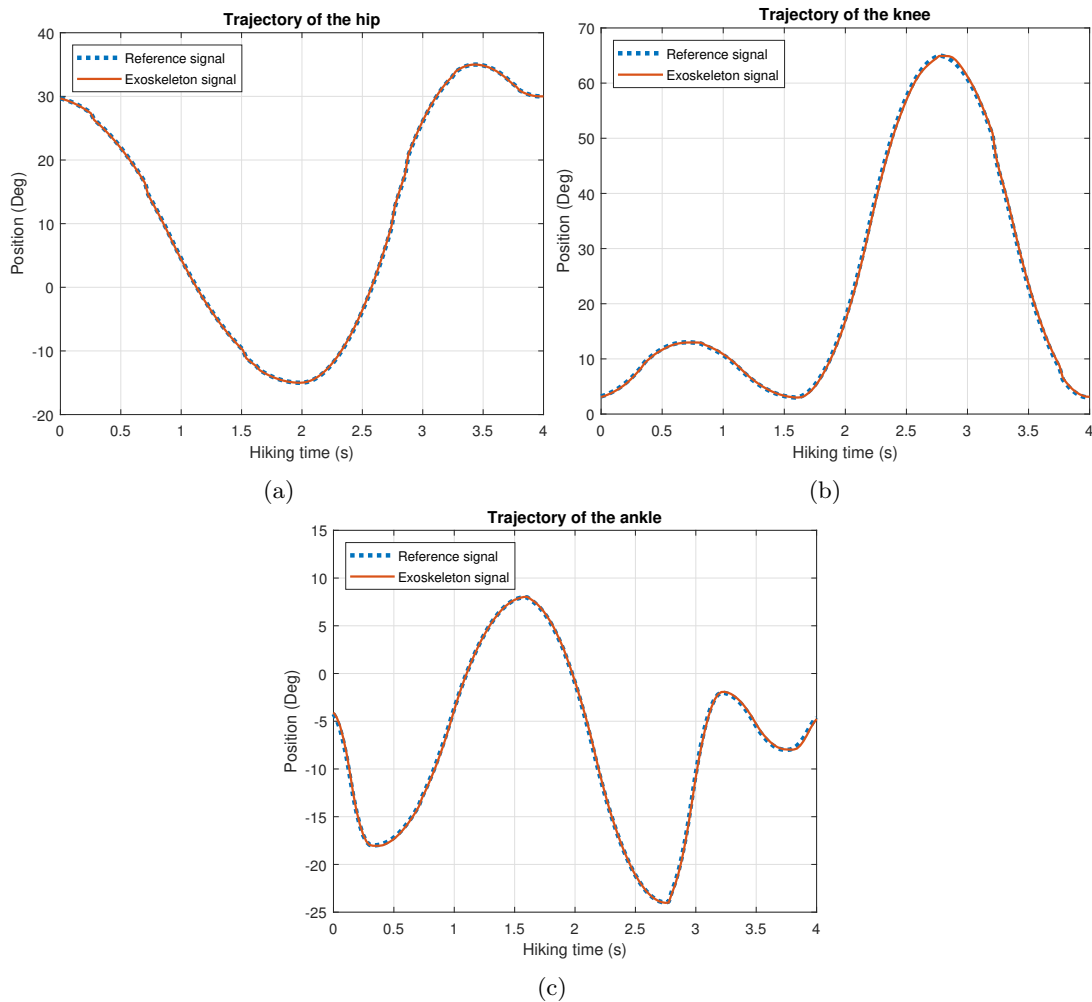


Figura 5.10: Trayectorias de caminata con 4 s y con una carga total de 15 kg



De la misma manera que se explicó en las pruebas sin carga, se puede ver que con peso y con un ciclo relativamente rápido de 4 s se pierde la precisión del control **PID** implementado dentro de los **EPOS4**. Los errores son más notables pero no afecta significativamente el funcionamiento del exoesqueleto, ya que el control actúa rápidamente para corregir los errores y llevar a la articulación a la referencia deseada, haciendo que el error sea casi imperceptible.

#### 5.1.4. Evaluación del consumo energético y torque aplicado

Por otra parte, ligado al funcionamiento de los reductores y a las características de los motores, provienen el consumo de corriente y el torque producido para accionar los actuadores. En las Tablas 5.1 y 5.2, se pueden observar las corrientes suministradas al motor cuando el prototipo está al vacío y luego con carga. Como era de esperarse, el consumo es mayor cuando el prototipo está con carga, teniendo incrementos promedio del 29 % para la cadera, 113 % para la rodilla y 62 % para el tobillo. Además, cuando el tiempo de marcha es menor (ciclo rápido) la corriente medida es mayor, teniendo incrementos promedio del 42 % para la cadera y 9 % para el tobillo; siendo lo contrario para el caso de la rodilla (incremento del 12 % en ciclo lento), que debido a características de su trayectoria permanece más tiempo ejecutando su trabajo con amplitudes mayores que las otras articulaciones. Finalmente, a pesar de que la cadera debe sostener el peso de toda la estructura, se puede ver que el consumo en esta articulación es relativamente bajo gracias al reductor y motor empleados en esta parte.

Los valores mostrados en las siguientes tablas de consumo de corriente y par aplicado en cada articulación, son una media de 10 experimentos, los mismos que fueron ejecutados en conjunto con las pruebas de control del prototipo.

Tabla 5.1: Consumo de corriente del exoesqueleto sin carga durante un ciclo marcha.

	4 segundos (mA)			10 segundos (mA)		
	Max	Avg	Min	Max	Avg	Min
<b>Cadera</b>	3021	889	1	1383	455	1
<b>Rodilla</b>	3479	1041	2	4427	993	1
<b>Tobillo</b>	969	310	5	560	232	1

Tabla 5.2: Consumo de corriente del exoesqueleto con carga durante un ciclo marcha.

	4 segundos (mA)			10 segundos (mA)		
	Max	Avg	Min	Max	Avg	Min
<b>Cadera</b>	3173	943	22	2312	634	11
<b>Rodilla</b>	6990	2281	1	8099	2498	2
<b>Tobillo</b>	1289	421	2	1284	356	1

En relación al par motor que ejercen las articulaciones, es en la cadera donde se produce

mayor momento de fuerza debido a la relación de transmisión de los reductores y a las características del actuador aquí instalado. Mientras que en las otras articulaciones se observa un comportamiento similar al de consumo de corriente. En las Tablas 5.3 y 5.4 se puede apreciar el comportamiento de cada una de las articulaciones.

Tabla 5.3: Torque del exoesqueleto sin carga durante un ciclo marcha.

	4 segundos (Nm)			10 segundos (Nm)		
	Max	Avg	Min	Max	Avg	Min
<b>Cadera</b>	41.74	0.01	13.67	19.11	0.01	6.3
<b>Rodilla</b>	5.49	0.003	1.64	6.99	0.002	1.57
<b>Tobillo</b>	1.53	0.008	0.49	0.88	0.002	0.37

Tabla 5.4: Torque del exoesqueleto con carga durante un ciclo marcha.

	4 segundos (Nm)			10 segundos (Nm)		
	Max	Avg	Min	Max	Avg	Min
<b>Cadera</b>	43.84	0.3	13.03	31.95	0.16	8.76
<b>Rodilla</b>	11.04	0.002	3.6	12.79	0.003	3.95
<b>Tobillo</b>	2.04	0.003	0.67	2.03	0	0.56

De las evaluaciones descritas, se puede concluir en primera instancia que las tres articulaciones tienen un control bastante preciso, pero es necesario resaltar el funcionamiento de la articulación de la cadera, la misma que no presenta variación alguna con respecto a la referencia en todas las pruebas realizadas. Con base en los resultados que se han alcanzado, se debe destacar el trabajo de los reductores con los que se está operando, en especial el de la cadera. Este reductor es de tipo cicloidal y basa su movimiento en un eje excéntrico, teniendo un rodamiento progresivo, parejo y sutil en los casquillos internos del reductor. Gracias a esta sencilla solución se logra balancear la carga para tener menor estrés y fricción entre los lóbulos y casquillos del reductor. Al tener un desgaste mínimo, la eficiencia del motor aumenta (87 % en este caso), lo cual significa un inminente ahorro energético. Sin lugar a duda, estas son las principales diferencias con un reductor planetario que hacen que el reductor cicloidal sea recomendado para este tipo de aplicaciones en las que los reductores están en movimiento generando estrés entre los engranes internos.

## 5.2. Evaluación del bus de comunicación

La comunicación se realiza mediante un bus **CAN**, al cual se encuentran conectados los **EPOS4** de cada articulación (dispositivos esclavos) y la Raspberry Pi 3 (dispositivo maestro). Las tramas de los mensajes **CAN** son generados automáticamente por las funciones de los dispositivos esclavos utilizando el protocolo **SDO**, descrito en la Sección 2.5.4. La Raspberry Pi 3 envía dos tipos de mensajes a las articulaciones: mensajes para mover las articulaciones y mensajes para capturar información (posición, velocidad y corriente) de las articulaciones. Las funciones *SetPositionProfile* y *MoveToPosition* se utilizan para generar el movimiento de los motores, en tanto que las funciones *GetPositionIs*, *GetVelocityIs* y *GetCurrentIs* se usan para capturar la información de las articulaciones.

Con el fin de comprobar que efectivamente se están enviando los mensajes con el protocolo **SDO**, se realiza la captura de los mensajes mediante *wireshark*. Con esto, es posible evaluar el tiempo que toma la transmisión de los mensajes por el bus **CAN** y el tiempo de procesamiento de los mismos en el dispositivo maestro. Finalmente, se presenta el número de muestras capturadas por la Raspberry Pi 3 al ejecutar movimientos del exoesqueleto con distintos tiempos de ciclo de caminata.

### 5.2.1. Mensajes CAN

Como se mencionó anteriormente, para comprobar que se están utilizando mensajes **CAN** con el protocolo **SDO** en la comunicación entre la Raspberry Pi 3 con cada uno de los **EPOS4**, se realiza una captura de paquetes del bus de comunicación con la herramienta *wireshark*. A manera de ilustración, en la Figura 5.11 se puede ver la captura de un mensaje que genera la función *GetPositionIs* enviada al dispositivo de la rodilla (nodo con el identificador número 2). Las funciones del **EPOS4** que capturan información de las articulaciones, envían dos mensajes desde el dispositivo maestro hacia el esclavo correspondiente, y se recibe un único mensaje de regreso, teniendo un total de 3 mensajes en la comunicación.

No.	Time	Src	Des	Protocol	Length	Info
1	0.000000000			CAN	32	STD: 0x00000002 40 64 60 00 00 00 00 00
2	0.000353907			CAN	32	STD: 0x00000002 40 64 60 00 00 00 00 00
3	0.000847136			CAN	32	STD: 0x00000582 43 64 60 00 9a 3a 00 00

Figura 5.11: Captura de mensaje SDO con *wireshark*

En los mensajes capturados de la Figura 5.11, se puede ver por su identificador que dos mensajes son enviados desde la Raspberry Pi 3 hacia el **EPOS4** de la rodilla (identificador 0x602h), y un mensaje es enviado como respuesta desde la rodilla al dispositivo maestro (identificador 0x502h). Además, se puede verificar que en realidad los mensajes enviados tienen una longitud

de 8 bytes. El resumen de los mensajes transmitidos sobre el bus CAN se los presenta en las Tablas 5.5 y 5.6

Tabla 5.5: Mensaje enviado desde la Raspberry Pi 3 hacia la rodilla

	Bytes enviados	Análisis
<i>Command Specifier</i>	0x40h (0x01000000b)	Indica que se desea leer un dato del diccionario de objetos
<i>Index</i>	0x6460h	El valor del índice del diccionario que se desea acceder es 0x6064 (valor de posición actual)
<i>Subindex</i>	0x00h	El valor del subíndice que se desea acceder es 0x00h
<i>Data</i>	0x00000000h	No se envía ningún dato, ya que únicamente se está consultando

Tabla 5.6: Mensaje enviado desde la rodilla hacia la Raspberry Pi 3

	Bytes enviados	Análisis
<i>Command Specifier</i>	0x43h (0x01000011b)	Indica que un dato fue leído del diccionario de objetos
<i>Index</i>	0x6460h	El valor del índice del diccionario que se va a acceder es 0x6064h (valor de posición actual)
<i>Subindex</i>	0x00h	El valor del subíndice que se accede es 0x00h
<i>Data</i>	0x9A3A0000h	El dato leído tiene un valor de 0x15002d, que pertenece a la posición de la rodilla

Efectivamente se puede ver que son mensajes que utilizan el protocolo SDO y que cumplen el formato explicado en la Sección 2.5.4. Todos los mensajes generados por las funciones de los EPOS4 tienen el mismo formato, pero varían los valores de los diccionarios que se acceden, los identificadores, que dependen de a qué articulación se envíe los mensajes, y los datos que se transmiten. Para los casos en los que se desea mover un motor, se envía información en el campo datos de la Raspberry Pi 3 a la articulación correspondiente, y el mensaje de respuesta no contiene información en el campo de datos. Lo opuesto sucede cuando se desea leer una variable (posición, velocidad o corriente) de alguna articulación. Es decir, la Raspberry Pi 3 envía mensaje con el campo de datos vacío hacia la articulación y ésta le responde con un nuevo

mensaje que contiene el valor requerido dentro del campo datos.

### 5.2.2. Análisis de los retardos generados en el envío de mensajes CAN

En esta sección se presenta el análisis de los retardos obtenidos al enviar todos los mensajes necesarios para el funcionamiento del exoesqueleto sobre el bus CAN.

Se realiza una comparación entre el tiempo que toma la transmisión de un mensaje CAN en el bus de comunicación, y el tiempo que toma el procesamiento del mismo en la Raspberry Pi 3. El tiempo de transmisión sobre el bus CAN también se lo obtiene a través de una captura de paquetes con *wireshark*. Por su parte, el tiempo de procesamiento se lo obtiene con un cronómetro colocado dentro de la programación que ejecuta las funciones de movimiento y captura de datos de los motores del exoesqueleto. Se miden los tiempos tanto de las funciones que envían mensajes para capturar información de las articulaciones, como de las funciones que envían mensajes para realizar el movimientos de las articulaciones.

En la Figura 5.12 se presenta una captura de los mensajes que envían las funciones *SetPositionProfile* y *MoveToPosition*. Estas funciones permiten cargar los parámetros para el movimiento de la articulación y mover la articulación respectivamente, en un tramo específico de la trayectoria de marcha. Al ejecutar las dos funciones se obtiene un total de 27 mensajes enviados por el bus de comunicaciones, los cuales se dividen en grupos de 3. En el caso de la Figura 5.12, los dos primeros de cada grupo son los mensajes enviados desde la Raspberry Pi 3 hacia el nodo 1 (articulación de la cadera), y el último es el mensaje de confirmación enviado por el nodo hacia la Raspberry Pi 3. En la Tabla 5.7 se presenta el tiempo que toma en transmitir, por el bus CAN, todos los 27 mensajes que realizan el movimiento de los motores. Las funciones que se utilizan para capturar información de las articulaciones, únicamente envían tres mensajes como se vio en la Figura 5.11. Se realizaron 10 capturas de información de una sola articulación y el promedio de tiempo que tarda en transmitir por el bus CAN y procesar en la Raspberry Pi, se presentan en la Tabla 5.7

Tabla 5.7: Tiempo de transmisión sobre el bus CAN en comparación con el tiempo de procesamiento de la Raspberry Pi 3

Tipo de mensaje	Tiempo en el bus CAN ( $\mu$ s)	Tiempo de procesamiento (ms)
Mover articulaciones	636.453	32.18489
Capturar datos	847.136	2.948583

Se puede ver que la transmisión de mensajes sobre el bus de comunicaciones están en el orden de los microsegundos. La velocidad de respuesta de cada uno de los EPOS4 es casi inmediata. Sin embargo, a pesar de la rápida transmisión de mensajes en el bus CAN, no se tiene una comunicación óptima. Esto se debe fundamentalmente a que los tiempos de procesamiento

No.	Time	Sou	Des	Protocol	Length	Info
1	0.000000000			CAN	32	STD: 0x000000601
2	0.000357586			CAN	32	STD: 0x000000601
3	0.000827197			CAN	32	STD: 0x000000581
4	0.002262695			CAN	32	STD: 0x000000601
5	0.002534139			CAN	32	STD: 0x000000601
6	0.002970158			CAN	32	STD: 0x000000581
7	0.004375971			CAN	32	STD: 0x000000601
8	0.004646998			CAN	32	STD: 0x000000601
9	0.005013281			CAN	32	STD: 0x000000581
10	0.007046092			CAN	32	STD: 0x000000601
11	0.007281601			CAN	32	STD: 0x000000601
12	0.007771992			CAN	32	STD: 0x000000581
13	0.009251030			CAN	32	STD: 0x000000601
14	0.009491018			CAN	32	STD: 0x000000601
15	0.009852926			CAN	32	STD: 0x000000581
16	0.011271081			CAN	32	STD: 0x000000601
17	0.011510236			CAN	32	STD: 0x000000601
18	0.011832302			CAN	32	STD: 0x000000581
19	0.013253426			CAN	32	STD: 0x000000601
20	0.013491643			CAN	32	STD: 0x000000601
21	0.013853187			CAN	32	STD: 0x000000581
22	0.026974433			CAN	32	STD: 0x000000601
23	0.027286969			CAN	32	STD: 0x000000601
24	0.027756111			CAN	32	STD: 0x000000581
25	0.029262856			CAN	32	STD: 0x000000601
26	0.029506333			CAN	32	STD: 0x000000601
27	0.029856887			CAN	32	STD: 0x000000581

Figura 5.12: Captura de mensaje SDO con *wireshark*

dentro de la Raspberry Pi 3 son muy elevados. Al ser tiempos elevados, no se puede capturar un número significativo de datos de cada una de las articulaciones. Además, en ciertos casos los mensajes de las funciones de los [EPOS4](#) no se envían a los instantes que deben ser, porque algún mensaje anterior se está procesando y el nuevo mensaje se encola en la Raspberry Pi 3 hasta que se pueda enviar. Esto causa desincronización en el movimiento de las articulaciones del exoesqueleto. Otra consecuencia es que puede no llegarse a capturar los datos en el instante de tiempo adecuado, perdiendo así la continuidad de los mismos durante el funcionamiento del exoesqueleto. En general, ciertas deformaciones observadas en las Figuras de la Sección 5.1 se deben precisamente a la pérdida de datos en esos instantes de tiempo.

Una posible solución a este problema es el cambiar el protocolo de mensajes [CAN](#) al protocolo de objeto de datos de procesos ([PDO](#), por sus siglas en inglés) y utilizar un dispositivo maestro que trabaje con un sistema operativo en tiempo real. Así se logrará que el procesamiento de los mensajes enviados y recibidos en el bus [CAN](#) tomen tiempos muy pequeños. Con ello se permitirá el envío de mensajes tanto de movimiento de las articulaciones como para la captura de información relevante de las mismas simultáneamente. Con estos cambios, el procesamiento será casi inmediato y no se encolarán los mensajes que se envían por el bus [CAN](#) dentro del dispositivo maestro. Además, se obtendrá mayor cantidad de datos capturados, ya que se puede capturar datos uno inmediatamente después de otro.

### 5.2.3. Cantidad de mensajes capturados

Para el funcionamiento del exoesqueleto se necesita enviar dos tipos de mensajes en paralelo. Los primeros mensajes se utilizan para generar el movimientos a los motores de cada articulación. El segundo tipo de mensajes se utiliza para capturar información relevante de cada articulación. Bajo la configuración actual, cada uno de los mensajes enviados por las funciones del EPOS4 tiene un alto tiempo de procesamiento dentro de la Raspberry Pi 3, como se indicó en la Sección 5.2.2, así que no es posible enviarlos de manera simultánea.

Los mensajes que se envían al bus CAN desde la Raspberry Pi 3 para recolectar la información de las articulaciones se encolan hasta que ésta pueda procesarlos, causando una demora en la captura de información. Como solución paliativa, se ha ingresado un retardo de tiempo extra antes de enviar los mensajes al bus de comunicaciones, caso contrario se enviarían mensajes uno después de otro causando una saturación en la comunicación entre la Raspberry Pi 3 y el bus CAN. Lo último generaría encolamiento de los mensajes que realizan el movimiento de los motores y la desincronización del movimiento del exoesqueleto. A pesar de esta limitación, se ha logrado obtener un número suficiente de muestras que permite mostrar las gráficas de la Sección 5.1, sin perder la continuidad del movimiento en cada una de la articulaciones.

En esta sección se realizan 10 pruebas, enviando mensajes que mueven todas las articulaciones del exoesqueleto. Además, se realiza la captura de los datos de posición de cada articulación. Se ejecuta el movimiento del exoesqueleto con tiempos desde los 4 segundos hasta los 10 segundos en el ciclo de caminata. En la Figura 5.13 se presenta un gráfico con la cantidad promedio de datos obtenidos en la Raspberry Pi 3 en cada uno de los tiempos de caminata.

Se puede ver que mientras mayor es el tiempo de caminata, se obtiene una cantidad mayor de muestras. Esto porque se dispone de mayor tiempo y por consiguiente se puede seguir enviando mensajes de captura de datos por más tiempo. A pesar que con el tiempo de 10 segundos en el ciclo de caminata se obtiene el mayor número de muestras, los datos no siempre son de mucha utilidad, ya que existen períodos de tiempo en los que no se tiene capturada la información. En los instantes que se está enviando información para mover las articulaciones, no se puede obtener la información de corrientes y velocidades. Éstos serían puntos claves donde se requiere dichos datos y no siempre se los puede obtener. Se puede realizar gráficas para analizar el comportamiento de las articulaciones del exoesqueleto, pero no dispone del número ideal de muestras. En la mayoría de los casos que se captura información se pierden los datos que debían ser capturados en los instantes de tiempo que se envían los mensajes de movimiento hacia los motores. Esto puede ocasionar una incongruencia en los datos capturados luego de cada ciclo de caminata del exoesqueleto. Además, la cantidad de datos capturados en cada prueba es variable, es decir, nunca se tiene la misma cantidad de muestras en todas las pruebas realizadas, por ello fue necesario realizar un promedio de todas las pruebas. Todo es una consecuencia directa del tiempo que le toma a la Raspberry Pi 3 el procesar los mensajes recibidos. En cada prueba, la

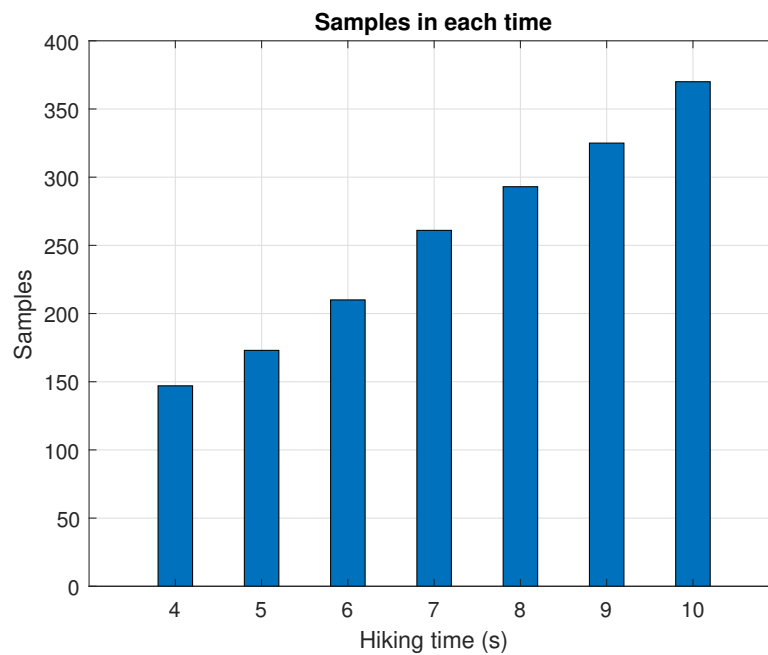


Figura 5.13: Número de muestras con cada tiempo de caminata

Raspberry Pi 3 puede tener más o menos procesos ejecutando al mismo tiempo que se captura la información y esto causa que algunos paquetes se pierdan. En un trabajo futuro se debe optar por cambiar el protocolo de los mensajes [CAN](#) por el protocolo [PDO](#), que envía mensajes sin confirmación con lo que se puede ganar más tiempo en su envío de mensajes. Además, se debería utilizar un dispositivo maestro que funcione con un sistema operativo en tiempo real, que permita realizar el procesamiento de los mensajes recibidos en él mismo de una manera inmediata.



## Capítulo 6

# Conclusiones y Recomendaciones

En este capítulo se presentan las conclusiones finales del trabajo realizado así como la interpretación de los resultados, las limitaciones que se encontraron en el proceso, y finalmente, se proponen maneras de ampliar y continuar con la investigación a futuro.

### 6.1. Conclusiones

Debido a que en el desarrollo de exoesqueletos se involucran muchas ramas de la ingeniería electrónica como el sistema de comunicación, la instrumentación (sensores y actuadores), el control y energía, además de las ramas de la mecánica y la física, este tipo de proyectos se convierten en sistemas completos de conocimiento, enriquecidos con teorías y técnicas que abarcan un marco muy grande dentro de la investigación, involucrándose además con campos enfocados en otros conocimientos, como el área de la salud.

En el desarrollo de exoesqueletos es imprescindible realizar estudios a un gran número de personas con y sin patologías motrices para que, con base en esto, se propongan opciones que permitan mejorar su condición de vida. En este caso específico, es primordial obtener la curva característica de un ciclo de marcha para que el prototipo robótico replique este movimiento y se logre una rehabilitación eficiente del paciente.

Para reproducir la marcha, el controlador del motor realiza una interpolación parabólica

basada en los puntos máximos y mínimos de la curva de referencia inicial, en conjunto con las velocidades y aceleraciones angulares calculadas para un tiempo específico. De esta manera, el motor y el controlador llegan a recrear un movimiento suavizado continuo que evita movimientos bruscos y favorece a los pacientes.

Los actuadores utilizados, principalmente los motores con su respectivo controlador, son equipos especializados en el desarrollo de exoesqueletos, por lo que los resultados que se han obtenido fueron satisfactorios, tanto al momento de realizar la comunicación, como al efectuar el control de las tres articulaciones del prototipo. El control **PID** en el que se basa es bastante robusto, llegando a recuperarse inmediatamente frente a perturbaciones. La comunicación para este sistema se fundamenta sobre el protocolo **CAN**, el cual es fácil y confiable de implementar gracias a su método de control de acceso al medio basado en detección de portadora y resolución de colisiones por prioridad; además de contar con una tasa de transmisión de 1 **Mbps**.

En cuanto a la estructura mecánica, los sistemas de reducción construidos tienen un juego mecánico inferior a  $2^\circ$ , lo cual satisface las necesidades de movimiento del prototipo, sin afectar a las mediciones de posición angular indicadas por los *encoders*. El sistema de barras deslizables permite ajustar el prototipo a la estatura del sujeto, y presenta la rigidez suficiente para soportar su masa.

El hecho de no poseer la madurez tecnológica con la que cuentan los países desarrollados y sus industrias, no debería considerarse como un limitante al momento de desarrollar aplicaciones complejas como ésta. Se puede adquirir o desarrollar herramientas tanto de software como de hardware que permitan potencializar las funciones de la tecnología con la que se cuenta.

Con esta primera versión del exoesqueleto de extremidades inferiores, únicamente se pudo realizar pruebas al vacío y con pesos relativamente pequeños para corroborar el correcto funcionamiento de los sistemas de control y de comunicaciones. Las pruebas no se lograron realizar con pacientes reales porque la estructura del exoesqueleto tiene implementado reductores de baja relación de reducción en las articulaciones de la rodilla y tobillo. Estos reductores hacen que la corriente en estas articulaciones se incremente a valores que exceden el valor nominal (cuando se usan cargas/pesos de relación similar al peso de una persona) de cada motor y desconecta de manera inmediata la fuente de alimentación del exoesqueleto por protección.

## 6.2. Limitaciones

Debido al tamaño del reductor utilizado en la cadera no se pueden realizar movimientos con ciclos de caminata menores a 4 segundos. La velocidad de giro de cada uno de los motores del exoesqueleto dependen del tiempo de ciclo de caminata y del reductor utilizado. Al realizar los cálculos de la velocidad a la que debe girar el motor de la cadera, considerando un tiempo de caminata menor a 4 segundos y la ganancia de 196:1 de su reductor, se obtienen velocidades

mayores a 3000 **rpm**. Este valor excede la velocidad nominal de 2500 **rpm** que tiene el motor. Se verificó que el exceso de velocidad produce un error en el controlador del motor y el exoesqueleto no se mueve. Por ello, únicamente se realizaron pruebas con ciclos de caminata mayores o iguales a 4 segundos.

El sistema operativo que utiliza la Raspberry Pi no funciona en tiempo real. Por este motivo, un mensaje **CAN** en el mejor de los casos toma un tiempo de 3 ms para ser procesado, pudiendo llegar hasta los 750 ms en ciertas pruebas. Adicionalmente, dicho tiempo de procesamiento es variable, y su variabilidad depende de los procesos que se estén ejecutando en su CPU. En muchos casos este tiempo es muy elevado y no le permite a la Raspberry Pi gestionar mensajes de control y lectura simultáneamente, causando que los mismos se encolen y no se capture información de realimentación en los tiempos indicados. Además, no se conoce con precisión la cantidad de datos que se van a obtener en la captura de información de cada prueba que se realice sobre el exoesqueleto.

El tiempo de procesamiento de la Raspberry Pi 3 ocasiona que en ciertas pruebas, los datos de posición no se capturen de manera periódica. Es por esto que algunas gráficas de las trayectorias de las articulaciones del exoesqueleto presentan pequeñas discontinuidades. Los motores se mueven acorde a la referencia indicada, pero los datos no se capturan de manera continua y se produce pequeñas deformidades en la gráfica de la trayectoria obtenida.

Durante la interconexión de los equipos, por un error humano, se obviaron los estándares con los que están formados los cables, causando el daño de los sensores por la sensibilidad que estos tienen. En este caso, se solventó el problema acoplado un *encoder* externo de igual precisión al motor mediante piezas 3D.

Al ser la primera versión del exoesqueleto, la evaluación final reveló que la estructura mecánica presenta algunas imperfecciones en su diseño. Resulta ser un diseño preliminar, causando que ciertas partes de la estructura pierdan sujeción con el uso. Esto, por su parte, ha ido generando mayor fricción de los actuadores colocados en la estructura, que a su vez se traduce en un incremento de la corriente de los motores. El exceso de corriente ocasionó incluso un daño en el motor de la cadera. La solución fue realizar nuevos orificios en los reductores de cada articulación para colocar pernos de sujeción y así evitar que las piezas del exoesqueleto dejen de ceder para prevenir nuevos daños de los motores.

### 6.3. Recomendaciones

Antes de utilizar los motores y controladores de los fabricantes *Maxon Motor*, se recomienda leer toda la documentación que los mismos fabricantes ponen a disposición de manera gratuita. Existe información sobre características técnicas de los equipos, las conexiones correctas que se debe realizar, configuraciones iniciales, instalación de librerías para la comunicación,

recomendaciones de diseños para placas de circuito impreso y funciones para utilizar los motores de manera sencilla. Es muy importante entender toda la información que se presenta en los documentos para utilizar los equipos de manera correcta, ya que alguna mala conexión o configuración no recomendada puede causar daños permanentes a los mismos.

Tener especial cuidado con los cables que se utilizan para conectar los actuadores y sensores hacia su respectivo controlador. Los cables de conexión poseen diferentes estándares, lo que significa que tendrán una diferente disposición en sus pines. Hay que tener presente esto para no tener inconvenientes al fabricar una PCB y sobretodo al momento de interconectar equipos. La omisión de estos estándares causaría el daño inmediato de los sensores por la sensibilidad y fragilidad que estos tienen.

## 6.4. Trabajos futuros

Con respecto al diseño mecánico, se debe considerar el aligeramiento del peso de los reductores, diseñando un sistema más compacto, que prescinda de bridas de acople entre el eje del actuador y el eje del reductor. También se debe aligerar el peso de los eslabones telescópicos que unen las articulaciones. Además, implementar reductores cicloidaes para las articulaciones de pie y tobillo, que reemplacen a los reductores epicicloides con una mayor relación de reducción. Finalmente, mejorar el sistema de sujeción del exoesqueleto al paciente, incluyendo almohadillas de foamy para reducir la presión ejercida por el exoesqueleto sobre la piel.

Para mejorar la comunicación entre los dispositivos del exoesqueleto, y sobre todo en lo referente a los tiempos de procesamiento, se debe utilizar un sistema operativo en tiempo real dentro del dispositivo maestro. Además, se debe evaluar el uso del protocolo PDO<sup>1</sup> en la red CAN. Se prevé que de esta manera se logrará reducir el tiempo de procesamiento de los mensajes CAN recibidos en el dispositivo maestro, y se enviará una menor cantidad de mensajes por cada acción que se desee realizar en el exoesqueleto. También se podrá presentar en tiempo real los datos capturados tanto de posición, velocidad y corrientes sin perder muestras durante el funcionamiento del exoesqueleto. El reducir los tiempos de procesamiento permitirá obtener una cantidad significativamente mayor de muestras de cada articulación que serán capturadas en tiempos específicos y periódicos.

Por otro lado, el funcionamiento del control PID *anti-windup* como se pudo ver es bastante preciso, pero es necesario probar estrategias de control inteligente para brindar características extras de autonomía, aprendizaje adaptativo y no linealidad. En relación al estudio de exoesqueletos, sería ideal utilizar estos algoritmos inteligentes para lograr un control complaciente, permitiendo que el prototipo se ajuste a las capacidades motoras del paciente.

---

<sup>1</sup>Datos en tiempo real enviados/recibidos por el controlador central, desde/hacia los diferentes nodos para cada período de muestra, basado en un modelo productor-consumidor.



# Anexos



## Apéndice A

# EPOS4 *Module* 50/8

Se presenta las características técnicas de la placa [EPOS4](#) utilizada en cada una de las articulaciones del exoesqueleto de extremidades inferiores.

Tabla A.1: Características eléctricas

Voltaje nominal de alimentación $+V_{cc}$	10 ... 50 VDC
Voltaje nominal lógico $+V_c$	10 ... 50 VDC
Voltajes de alimentación absolutos $+V_{min}/+V_{max}$	8 VDC/56 VDC
Voltaje de salida máximo	$0.9 \cdot +V_{cc}$
Corriente de salida $I_{con}/I_{max}$	8A/30A

Tabla A.2: Entradas y salidas

Entradas digitales 1, 2, 3, 4 (propósito general)	+2.1 ... +36 VDC
Salidas digitales 1, 2 (propósito general)	max. 36 VDC /I<500mA
Entradas STO 1, 2	+4.5 ... +30 VDC
Salida STO	max. 30 VDC/I<15mA
Entradas analógicas 1, 2	Resolución de 12 bits
Salidas analógicas 1, 2	-10 ... +10 VDC, Resolución de 12 bits, -4 ... +4 VDC, referenciados a GND
Señales digitales de sensores Hall H1, H2, H3	+2 ... +24 VDC
Encoder incremental digital A, A\, B, B\, I, I\	EIA RS422, max 6.25 MHz

Tabla A.3: Interfaces

USB 2.0 / USB 3.0	Máxima velocidad
RS232	max. 115'200 bits/s, Se necesita un transceptor externo
CAN	max. 1 Mbit/s
EtherCAT	Full duplex, 100 Mbit/s; IEEE 802.3 base 100 Tx

## A.1. Pines utilizados del EPOS4

Los pines listados a continuación son los que se utilizaron para el control de los motores que se colocan en cada una de las articulaciones del exoesqueleto.

- Pines A1 ... A4: Bobina 1 del motor [EC](#).
- Pines A5 ... A8: Bobina 2 del motor [EC](#).
- Pines A9 ... A12: Bobina 3 del motor [EC](#).
- Pines A13 ... A16: Voltaje de alimentación (10 ... 50 VDC).
- Pin A17: Voltaje lógico (10 ... 50 VDC).
- Pines 18 ... A22: Tierra.
- Pin A23: Sensor *hall* 1.
- Pin A24: Sensor *hall* 2.
- Pin A25: Sensor *hall* 3.
- Pin A26: Voltaje para sensores (+5 VDC).
- Pin A27: Canal A del *encoder* incremental digital.
- Pin A28: Complemento del canal A del *encoder* incremental digital.
- Pin A29: Canal B del *encoder* incremental digital.
- Pin A30: Complemento del canal B del *encoder* incremental digital.
- Pines B1 ... B4: Entradas digitales.
- Pines B5, B6: Salidas digitales.
- Pin B7: Canal A del *encoder* incremental digital externo.
- Pin B8: Complemento del canal A del *encoder* incremental digital externo.
- Pin B9: Canal B del *encoder* incremental digital externo.
- Pin B10: Complemento del canal B del *encoder* incremental digital externo.
- Pin B15: Voltaje auxiliar para sensores (+5 VDC).
- Pin B16: Tierra.
- Pines B17 ... B20: Entradas del [STO](#).
- Pines B21, B22: Salidas del [STO](#).
- Pines B23 ... B26: Entradas analógicas.



- Pines B27, B28: Salidas analógicas.
- Pin B35: Línea del bus **CAN High**.
- Pin B36: Línea del bus **CAN Low**.
- Pin B37: Tierra.

**NOTA:** Todas las tierras están cortocircuitadas.

La distribución de pines del **EPOS4** se presenta en la Figura A.1.

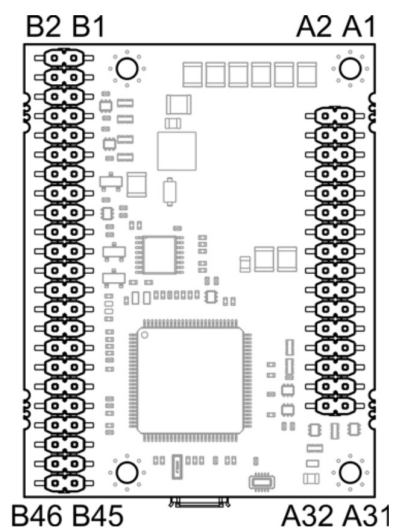


Figura A.1: Distribución de pines en el EPOS4



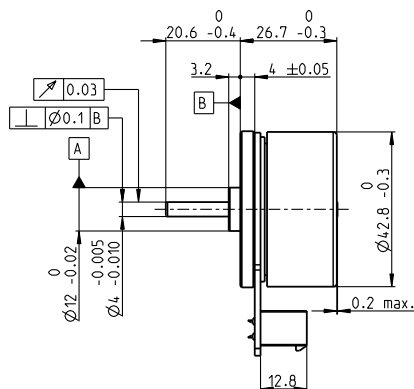
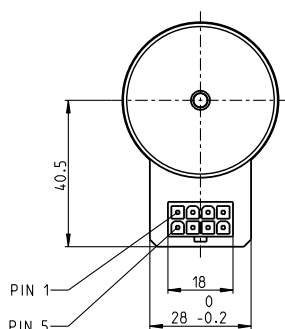


## Apéndice B

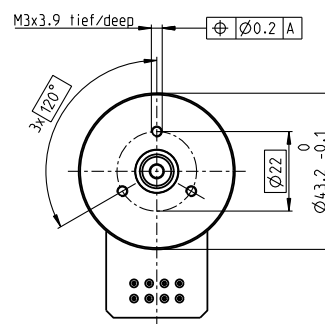
# Hojas de especificaciones técnicas de los motores EC 45 *flat* y EC 90 *flat* con *encoder* interno MILE

En esta sección se presenta las características técnicas de los motores utilizados en las articulaciones del exoesqueleto. Los motores EC 45 son para las articulaciones de la rodilla y tobillo los cuales son el modelo 397172 con *encoder* 462004, mientras que el motor EC 90 es para la cadera que es el modelo 323772 con *encoder* 453233. Las características se presentan a continuación.

# EC 45 flat Ø42.8 mm, brushless, 70 Watt



Connector:  
39-28-1083 Molex



M 1:2

- Stock program
- Standard program
- Special program (on request)

## Part Numbers

with Hall sensors

397172 402685 402686 402687

## Motor Data (provisional)

### Values at nominal voltage

1 Nominal voltage	V	24	30	36	48
2 No load speed	rpm	6110	6230	6330	3440
3 No load current	mA	234	194	166	48.1
4 Nominal speed	rpm	4860	4990	5080	2540
5 Nominal torque (max. continuous torque)	mNm	128	112	108	134
6 Nominal current (max. continuous current)	A	3.21	2.36	1.93	0.936
7 Stall torque	mNm	1460	1170	1100	915
8 Stall current	A	39.5	25.8	20.7	6.97
9 Max. efficiency	%	85	84	83	84

### Characteristics

10 Terminal resistance phase to phase	Ω	0.608	1.16	1.74	6.89
11 Terminal inductance phase to phase	mH	0.463	0.691	0.966	5.85
12 Torque constant	mNm / A	36.9	45.1	53.3	131
13 Speed constant	rpm / V	259	212	179	72.7
14 Speed / torque gradient	rpm / mNm	4.26	5.44	5.85	3.82
15 Mechanical time constant	ms	8.07	10.3	11.1	7.24
16 Rotor inertia	gcm <sup>2</sup>	181	181	181	181

## Specifications

### Thermal data

17 Thermal resistance housing-ambient	3.56 K/W
18 Thermal resistance winding-housing	4.1 K/W
19 Thermal time constant winding	29.6 s
20 Thermal time constant motor	178 s
21 Ambient temperature	-40 ... +100°C
22 Max. winding temperature	+125°C

### Mechanical data (preloaded ball bearings)

23 Max. speed	10 000 rpm
24 Axial play at axial load < 4.0 N	0 mm
> 4.0 N	0.14 mm
25 Radial play	preloaded
26 Max. axial load (dynamic)	3.8 N
27 Max. force for press fits (static)	50 N
(static, shaft supported)	1000 N
28 Max. radial load, 5 mm from flange	21 N

### Other specifications

29 Number of pole pairs	8
30 Number of phases	3
31 Weight of motor	141 g

Values listed in the table are nominal.

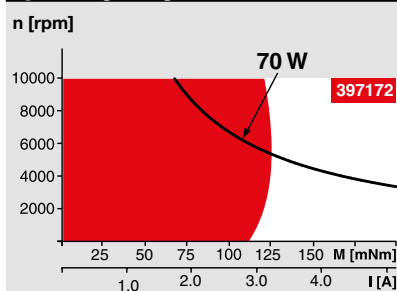
### Connection

Pin 1	Hall sensor 1*
Pin 2	Hall sensor 2*
Pin 3	V <sub>hall</sub> 4.5 ... 18 VDC
Pin 4	Motor winding 3
Pin 5	Hall sensor 3*
Pin 6	GND
Pin 7	Motor winding 1
Pin 8	Motor winding 2
*Internal pull-up (7 ... 13 kΩ) on pin 3	
Wiring diagram for Hall sensors see p. 43	

### Cable

Connection cable Universal, L = 500 mm	339380
Connection cable to EPOS, L = 500 mm	354045

## Operating Range



## Comments

- Continuous operation**  
In observation of above listed thermal resistance (lines 17 and 18) the maximum permissible winding temperature will be reached during continuous operation at 25°C ambient.  
= Thermal limit.
- Short term operation**  
The motor may be briefly overloaded (recurring).
- Assigned power rating**

## maxon Modular System

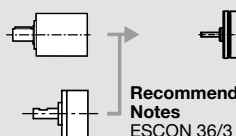
Overview on page 28-36

### Planetary Gearhead

Ø42 mm  
3 - 15 Nm  
Page 347

### Spur Gearhead

Ø45 mm  
0.5 - 2.0 Nm  
Page 349



### Recommended Electronics:

Notes	Page 32
ESCON 36/3 EC	427
ESCON Mod. 50/4 EC-S	427
ESCON Module 50/5	427
ESCON 50/5	428
DEC Module 50/5	430
EPOS2 Module 36/2	434
EPOS2 24/5, 50/5	435
EPOS2 P 24/5	438
EPOS4 Module/CB 50/5	442
MAXPOS 50/5	447

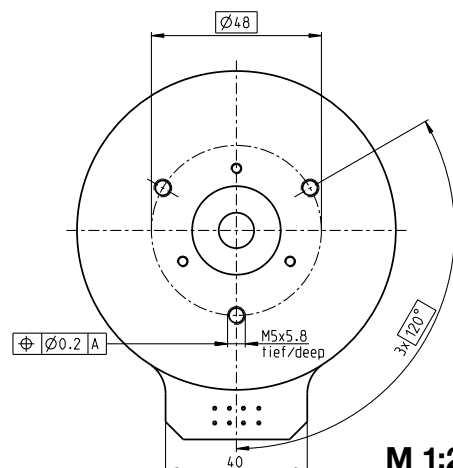
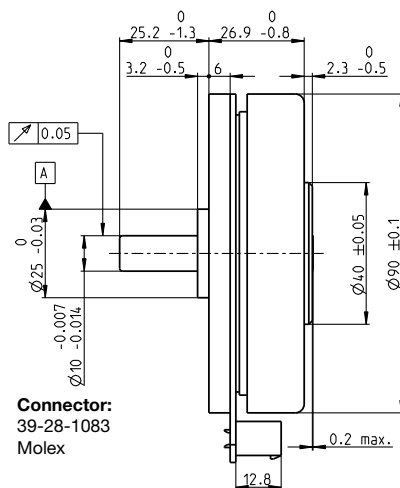
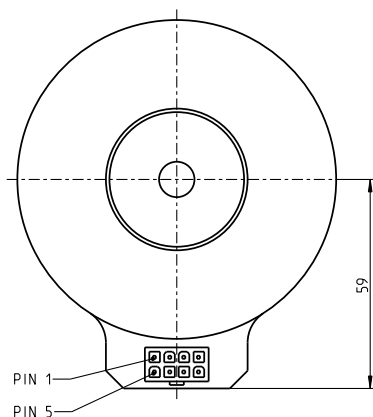
**Encoder MILE**  
256 - 2048 CPT,  
2 channels  
Page 388

### Option

With Cable and Connector  
(Ambient temperature -20 ... +100°C)

# EC 90 flat Ø90 mm, brushless, 90 Watt

maxon flat motor



- Stock program
- Standard program
- Special program (on request)

## Part Numbers

with Hall sensors

323772 429271 244879

## Motor Data

### Values at nominal voltage

1 Nominal voltage	V	24	36	48
2 No load speed	rpm	3190	3120	2080
3 No load current	mA	544	348	135
4 Nominal speed	rpm	2590	2510	1610
5 Nominal torque (max. continuous torque)	mNm	444	560	533
6 Nominal current (max. continuous current)	A	6.06	4.76	2.27
7 Stall torque	mNm	4940	7480	4570
8 Stall current	A	70	69	21.1
9 Max. efficiency	%	84	87	85

### Characteristics

10 Terminal resistance phase to phase	Ω	0.343	0.522	2.28
11 Terminal inductance phase to phase	mH	0.264	0.625	2.5
12 Torque constant	mNm/A	70.5	109	217
13 Speed constant	rpm/V	135	88	44
14 Speed/torque gradient	rpm/mNm	0.659	0.423	0.462
15 Mechanical time constant	ms	21.1	13.6	14.8
16 Rotor inertia	gcm <sup>2</sup>	3060	3060	3060

## Specifications

### Thermal data

17 Thermal resistance housing-ambient	1.91 K/W
18 Thermal resistance winding-housing	2.6 K/W
19 Thermal time constant winding	46 s
20 Thermal time constant motor	283 s
21 Ambient temperature	-40...+100°C
22 Max. winding temperature	+125°C

### Mechanical data (preloaded ball bearings)

23 Max. speed	5000 rpm
24 Axial play at axial load < 15 N	0 mm
> 15 N	0.14 mm
25 Radial play	preloaded
26 Max. axial load (dynamic)	12 N
27 Max. force for press fits (static)	183 N
(static, shaft supported)	8000 N
28 Max. radial load, 5 mm from flange	68 N

### Other specifications

29 Number of pole pairs	12
30 Number of phases	3
31 Weight of motor	600 g

Values listed in the table are nominal.

### Connection

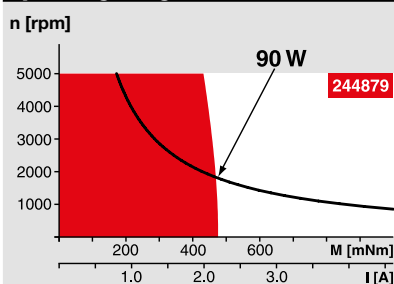
Pin 1	Hall sensor 1
Pin 2	Hall sensor 2
Pin 3	V <sub>Hall</sub> 4.5...18 VDC
Pin 4	Motor winding 3
Pin 5	Hall sensor 3
Pin 6	GND
Pin 7	Motor winding 1
Pin 8	Motor winding 2

Wiring diagram for Hall sensors see p. 43

### Cable

Connection cable Universal, L = 500 mm	339380
Connection cable to EPOS2, L = 500 mm	354045

## Operating Range



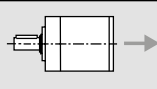
## Comments

- **Continuous operation**  
In observation of above listed thermal resistance (lines 17 and 18) the maximum permissible winding temperature will be reached during continuous operation at 25°C ambient.  
= Thermal limit.
- **Short term operation**  
The motor may be briefly overloaded (recurring).
- **Assigned power rating**

## maxon Modular System

Overview on page 28–36

**Planetary Gearhead**  
Ø52 mm  
4 - 30 Nm  
Page 351



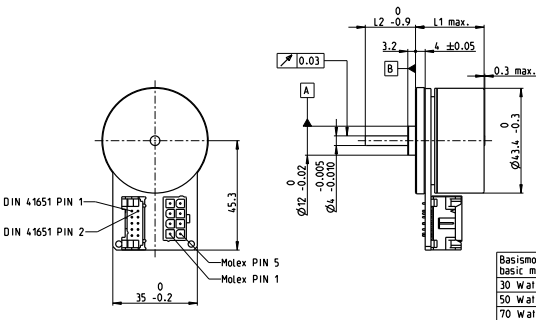
**Encoder MILE**  
512 - 6400 CPT,  
2 channels  
Page 390

## Recommended Electronics:

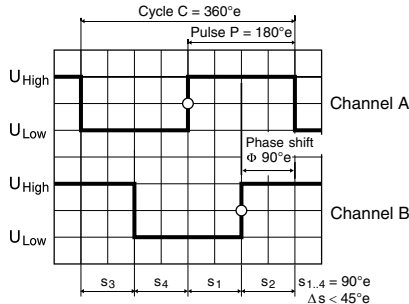
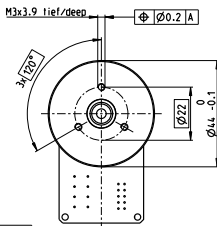
Notes	Page 32
ESCON Mod. 50/4 EC-S	427
ESCON Mod. 50/5	427
ESCON 50/5	428
ESCON 70/10	428
DEC Module 50/5	430
EPOS2 24/5, 50/5, 70/10	435
EPOS2 P 24/5	438
EPOS4 Module/CB 50/5	442
EPOS4 Module 50/8	443
EPOS4 Comp. 50/8 CAN	443
MAXPOS 50/5	447

# Encoder MILE 256–2048 CPT, 2 Channels, with Line Driver

Integrated into motor



Basic motor	L1 [mm]	L2 [mm]
30 Watt	19.4	20.2
50 Watt	22.6	20.2
70 Watt	28.4	20.8



M 1:3

Direction of rotation cw (definition cw p. 60)

- Stock program
- Standard program
- Special program (on request)

## Article Numbers

462002	462003	462004	462005
--------	--------	--------	--------

## Type

Counts per turn	256	512	1024	2048
Number of channels	2	2	2	2
Max. operating frequency (kHz)	500	500	500	500
Max. speed (rpm)	10 000	10 000	10 000	10 000



## maxon Modular System

+ Motor	Page	+ Gearhead	Page	+ Brake	Page	Overall length [mm] / see Gearhead				
EC 45 flat, 30 W, A	265					19.4	19.4	19.4	19.4	
EC 45 flat, 30 W, A	265	GP 42, 3 - 15 Nm	347							
EC 45 flat, 30 W, A	265	GS 45, 0.5 - 2.0 Nm	349							
EC 45 flat, 50 W, A	266					22.6	22.6	22.6	22.6	
EC 45 flat, 50 W, A	266	GP 42, 3 - 15 Nm	347							
EC 45 flat, 50 W, A	266	GS 45, 0.5 - 2.0 Nm	349							
EC 45 flat, 70 W, A	267					28.4	28.4	28.4	28.4	
EC 45 flat, 70 W, A	267	GP 42, 3 - 15 Nm	347							
EC 45 flat, 70 W, A	267	GS 45, 0.5 - 2.0 Nm	349							

## Technical Data

Supply voltage $V_{CC}$	5 B ± 10%
Typical current draw	14 mA
Output signal	CMOS compatible
State length $s_n$ , 90°e (1000 rpm)	45...135°e
Signal rise time	
(typically, at $C_L = 25$ pF, $R_L = 1$ kΩ, 25°C)	100 ns
Signal fall time	
(typically, at $C_L = 25$ pF, $R_L = 1$ kΩ, 25°C)	100 ns
Operating temperature range	-40...+100°C
Moment of inertia of code wheel	≤ 3.5 gcm <sup>2</sup>
Output current per channel	max. 4 mA
Open collector output of the Hall sensors with integrated pull-up resistor	10 kΩ ± 20%
Wiring diagram for Hall sensors see p. 43	

Additional information can be found under 'Downloads' in the maxon online shop.

## Pin Allocation

Connection motor	Connection Encoder
Pin 1 Hall sensor 1*	Pin 1 N.C.
Pin 2 Hall sensor 2*	Pin 2 $V_{CC}$
Pin 3 $V_{Hall}$ 4.5...18 VDC	Pin 3 GND
Pin 4 Motor winding 3	Pin 4 N.C.
Pin 5 Hall sensor 3*	Pin 5 Channel A
Pin 6 GND	Pin 6 Channel A
Pin 7 Motor winding 1	Pin 7 Channel B
Pin 8 Motor winding 2	Pin 8 Channel B
	Pin 9 Do not connect
	Pin 10 Do not connect

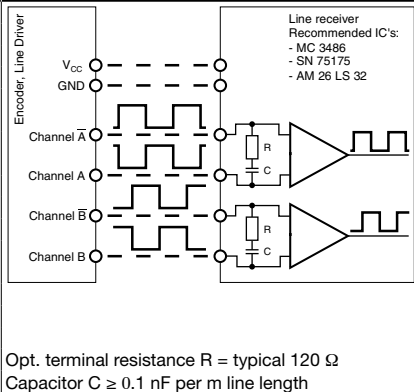
\*Internal pull-up (10 kΩ) on pin 3 ( $V_{Hall}$ )

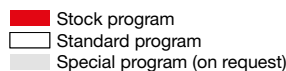
**Connector:**  
39-28-1083 Molex  
DIN 41651/EN 60603-13

**Note:** Pull-down resistors < 100 kΩ on the encoder outputs are not permitted. Pull-up resistors are permitted, but not required.

**Option**  
With Cable and Connector  
(Ambient temperature -20...+100°C)

## Pin Allocation





453234	409996	453233	411964	453232	411965	453231	411966
--------	--------	--------	--------	--------	--------	--------	--------

<b>+ Motor</b>	Page	<b>+ Gearhead</b>	Page	<b>+ Brake</b>	Page	<b>Overall length [mm] / ● see Gearhead</b>						
EC 90 flat	271					29.2	29.2	29.2	29.2	29.2	29.2	29.2
EC 90 flat	271	GP 52. 4 - 30 Nm	351			●	●	●	●	●	●	●

Supply voltage $V_{CC}$	$5 V \pm 10\%$
Typical current draw	14 mA
Output signal driver used:	EIA Standard RS422 AM26C31QD
State length $s_n$ (500 rpm)	$90^\circ e \pm 45^\circ e$
Signal rise and fall times (typically, at $C_L = 120$ pF, $R_L = 100 \Omega$ )	20 ns
Operating temperature range	$-40...+100^\circ C$
Moment of inertia of code wheel	$\leq 65$ gcm <sup>2</sup>
Output current per channel	min. -20 mA, max. 20 mA
Open collector output of the Hall sensors with integrated pull-up resistor	$10 k\Omega \pm 20\%$
Wiring diagram for Hall sensors see p. 43	

Pin 1	Hall sensor 1*
Pin 2	Hall sensor 2*
Pin 3	$V_{Hall}$ 4.5...18 VDC
Pin 4	Motor winding 3
Pin 5	Hall sensor 3*
Pin 6	GND
Pin 7	Motor winding 1
Pin 8	Motor winding 2

**Connector:**  
39-28-1083 Molex  
DIN 41651/EN 60603-13

Pin 1 N.C.  
Pin 2  $V_{CC}$   
Pin 3 GND  
Pin 4 N.C.  
Pin 5 Channel  $\bar{A}$   
Pin 6 Channel A  
Pin 7 Channel  $\bar{B}$   
Pin 8 Channel B  
Pin 9 Do not connect  
Pin 10 Do not connect

The diagram illustrates a 4-wire CAN bus system. On the left, an **Encoder, Line Driver, AM26C31D** is shown with four output lines:  $V_{CC}$ , GND, Channel  $\bar{A}$ , and Channel A. Channel  $\bar{B}$  and Channel B are also shown but are not connected to the bus. The bus lines are connected to two **Line receiver** units, each containing a recommended IC: MC 3486, SN 75175, or AM 26 LS 32. The receivers are connected to the bus lines through a resistor  $R$ . The timing diagram shows the signals for Channel  $\bar{A}$ , Channel A, Channel  $\bar{B}$ , and Channel B, and the resulting bus signals for the two receivers.

Opt. terminal resistance  $R = \text{typical } 120 \, \Omega$





## Apéndice C

# Conexión entre motores y los controladores EPOS4

### C.1. Distribución de pines y conexiones

En las Figuras C.1, C.2 y A.1 se presenta la distribución de pines de los motores utilizados en el exoesqueleto y el driver utilizado para controlar los mismos.

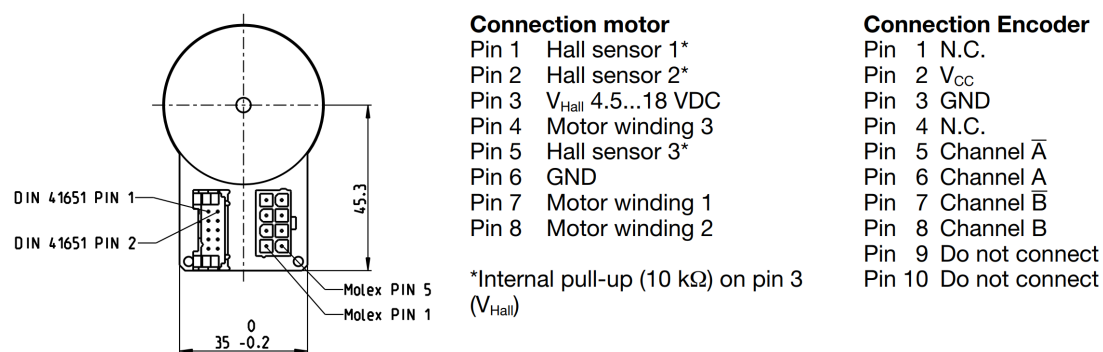


Figura C.1: Ubicación y distribución de pines de *Maxon Motor EC45* con *encoder*.

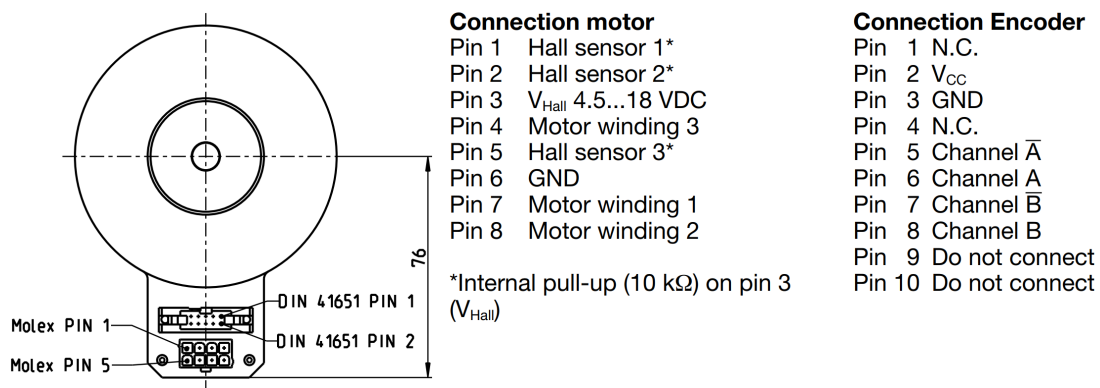


Figura C.2: Ubicación y distribución de pines de *Maxon Motor* EC90 con *encoder*.

Primero se necesita conectar la alimentación de los motores con sus respectivos devanados al [EPOS4](#) como se indica en la Tabla [C.1](#).

Tabla C.1: Conexión entre el motor y el controlador

Función	PIN motor	PIN controlador
VCC +10...+50 VDC	-	A13-A16
GND	6	A18-A22
LOGIC SUPPLY	-	A17
MOTOR WINDING 1	7	A1-A4
MOTOR WINDING 2	8	A5-A8
MOTOR WINDING 3	4	A9-A12
HALL SENSOR 1	1	A23
HALL SENSOR 2	2	A24
HALL SENSOR 3	5	A25
HALL SENSOR +5 VDC	3	A26

Los motores tienen *encoder* interno que también deben ser conectados al [EPOS4](#) como se indica en la Tabla [C.2](#).

Tabla C.2: Conexión entre el *encoder* y el controlador

Función	PIN encoder	PIN controlador
VCC +5v	2	A26
GND	3	A18-A22
CHANNEL A\	5	A28
CHANNEL A	6	A27
CHANNEL B\	7	A30
CHANNEL B	8	A29
Sin conectar	1, 4, 9, 10	-

En la articulación del tobillo se está utilizando un motor sin encoder, por lo que es necesario colocar un encoder externo de la misma resolución del los *encoders* internos de los motores *Maxon*. El *encoder* externo debe ir conectado en pines diferentes del [EPOS4](#), la conexión necesaria se presenta en la Tabla [C.3](#)

Tabla C.3: Conexión entre el *encoder* externo y el controlador

Función	PIN encoder	PIN controlador
VCC +5v	6	B15
GND	4	B16
CHANNEL A\	11	B8
CHANNEL A	10	B7
CHANNEL B\	9	B10
CHANNEL B	8	B9

Los controladores [EPOS4](#) tienen un sistema de seguridad para evitar sobre-torques que generan picos de corriente y podrían quemar los motores [STO](#). Lo que hace es desconectar los motores del [EPOS4](#). La implementación de esta seguridad es necesaria en la conexión de los motores con el driver. La conexión se presenta en la Tabla [C.4](#).

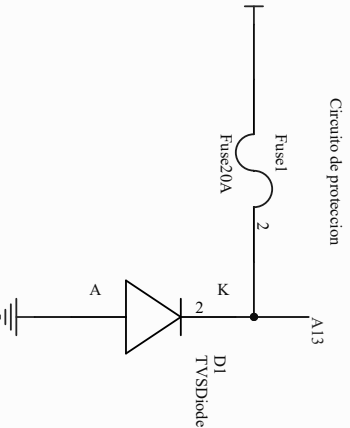
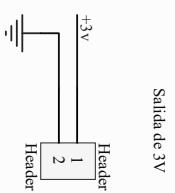
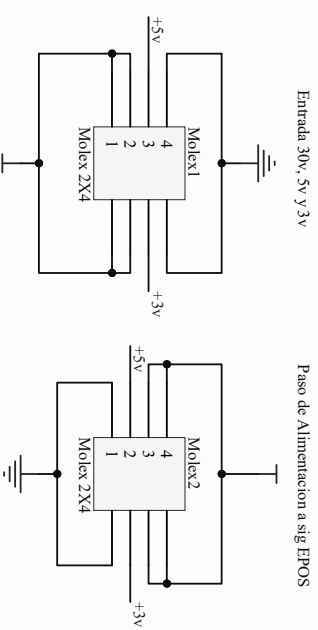
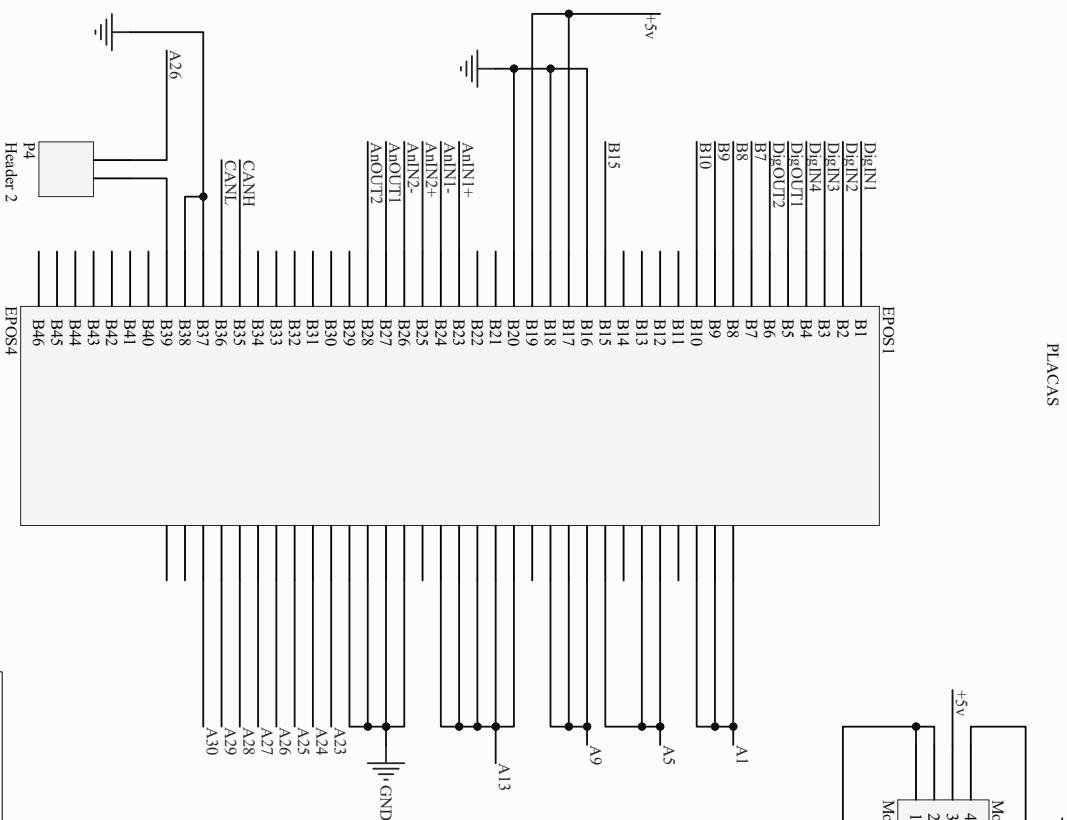
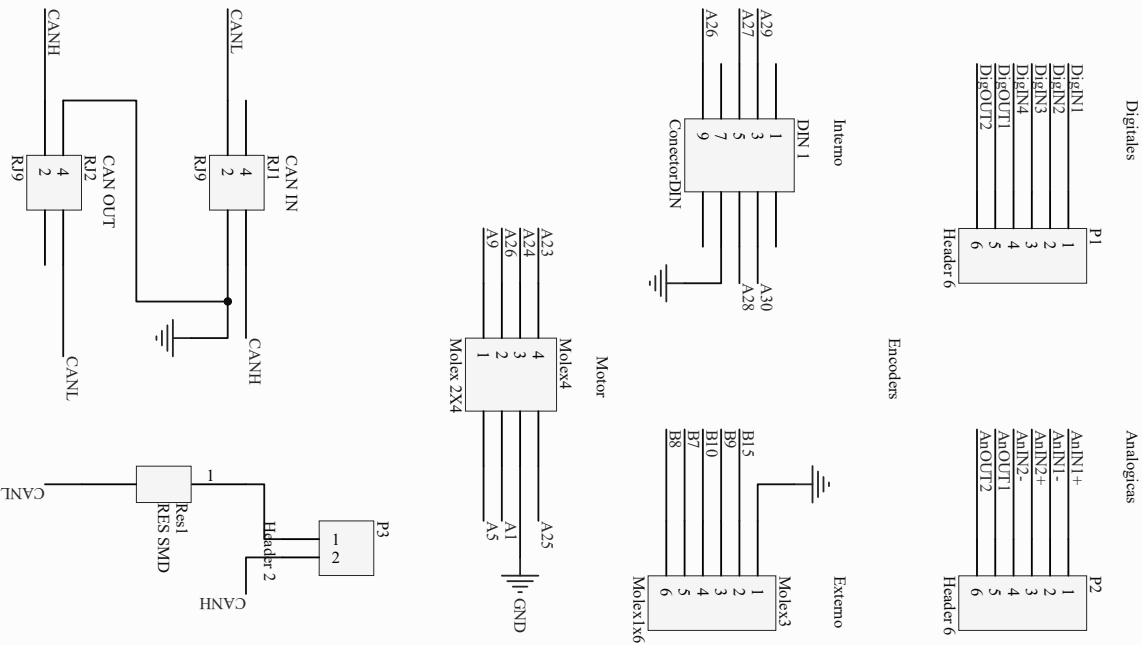
Tabla C.4: Conexión del [STO](#)

Función	PIN controlador	Descripción
STO-IN1+	B17	+5 VDC
STO-IN1-	B18	GND
STO-IN2+	B19	+5 VDC
STO-IN2-	B20	GND

## C.2. Diseño de placas de circuito impreso (PCB)

Para tener un orden en la conexión de cada uno de los controladores [EPOS4](#) hacia los motores de cada articulación y la red [CAN](#) se necesita diseñar [PCBs](#). Se diseña una sola placa que será utilizado para colocar el controlador [EPOS4](#) de cada una de las articulaciones del exoesqueleto (El mismo diseño para todas las articulaciones). La placa está compuesta por zócalos que permiten conectar el controlador. Además, está compuesta por una variedad de conectores, cada uno para una función específica ([CAN](#), *encoder*, motor).

Primero se necesita realizar el diseño esquemático de la placa, para lo cual se utiliza las conexiones indicadas en las Tablas [C.1](#), [C.2](#), [C.3](#) y [C.4](#). El diseño se realiza en el software *Altium Designer*, y se realiza una librería que contenga todos los elementos que se utilizan en las placas. El diseño esquemático que se utilizan en el exoesqueleto se presentan a continuación, el cual representa el circuito implementado en cada articulación del exoesqueleto. La placa está diseñada para que la alimentación ingrese por la parte superior de la misma, alimente el circuito de la articulación y por la parte inferior salga la misma alimentación hacia la siguiente placa (conexión en paralelo).



Title			Revision	
Size	Number			
A4				
Date:	30/12/2018	Sheet of		
File:	C:\Users\...PlacasSchDoc	Drawn By:		

Con el diseño esquemático, se procede a construir la placa de circuito impreso. Debido a la gran cantidad de conexiones en la placa, se genera una placa con conexiones a doble cara. El diseño final se presenta en la Figura C.3, donde se obtiene una placa con dimensiones de 100 mm X 75 mm. La placa cuenta con conectores RJ-9 para la conexión al bus CAN, conectores Molex 2x4 para la alimentación de los EPOS4 y motores, conectores DIN para el *encoder* interno, y conectores Moles 1X6 para *encoder* externo. Además, se ha colocado elementos de protección como un fusible y diodo en antiparalelo, una resistencia de 120 ohm con un *jumper* para cerrar el bus CAN y dos peinetas conectadas a las entradas analógicas y digitales del EPOS4 para utilizarlas en algún trabajo futuro.

Se fabrican tres placas con el mismo diseño, una para cada articulación, y vienen listas para colocarles tanto motores con *encoder* interno como externo, ya que dentro del exoesqueleto se dispone de los dos tipos. Solo se necesita colocar cada placa cerca de su articulación y mediante cables dedicados conectar todo el circuito hacia los motores y la red CAN. Para cerrar el bus de comunicaciones, en la última placa del exoesqueleto (ubicado en el tobillo) debe ser colocado el *jumper* para indicar que es el dispositivo terminal de la red.

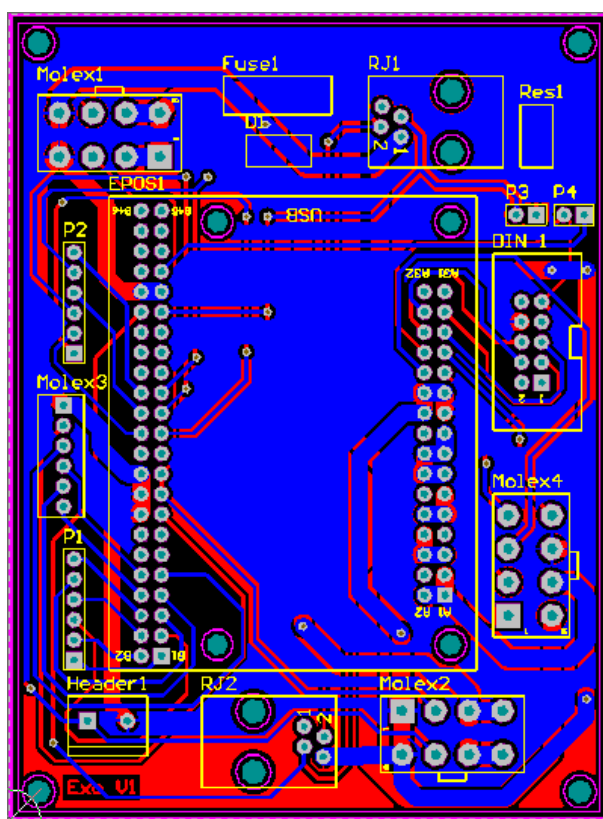


Figura C.3: Diseño de circuito impreso de placas del exoesqueleto

## Apéndice D

# Instalación de la librería EPOS\_Linux\_Library

Debido a que se va a utilizar una Raspberry PI 3, se necesita instalar la librería compatible para sistema operativo de Linux (Debian). La librería se la puede descargar de la página del fabricante mediante el siguiente enlace: [Librería Linux](#). (Downloads -Software/firmware)). (Véase Figura D.1)

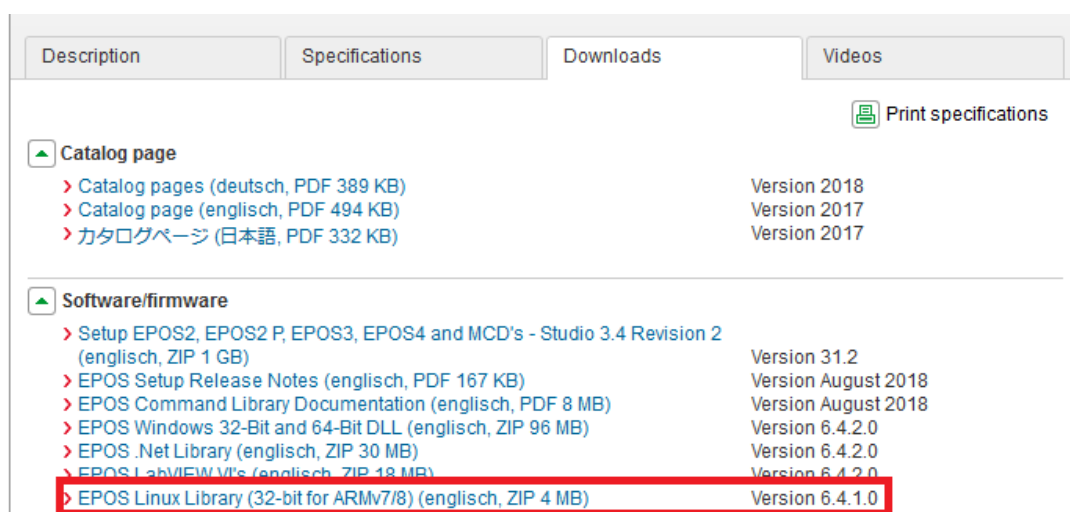


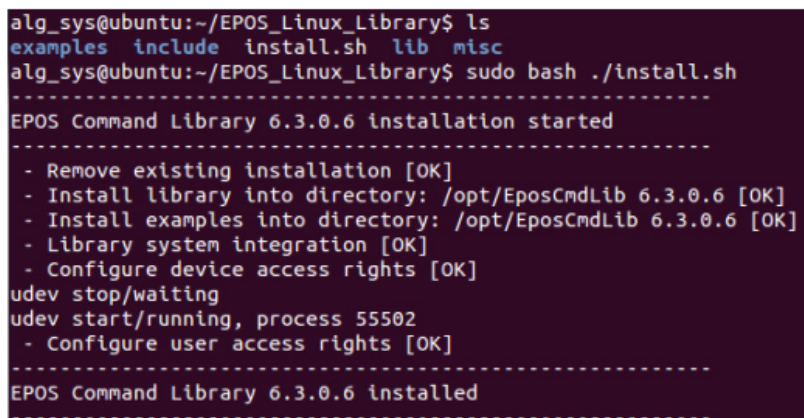
Figura D.1: Descarga de librería para EPOS4

La librería se descarga comprimida en formato .zip, así que será necesario descomprimir y luego acceder a la carpeta descomprimida “EPOS\_Linux\_Library” mediante la terminal de la Raspberry Pi 3:

```
cd EPOS_Linux_Library
```

Dentro de la carpeta se encuentra el archivo “install.sh” que será necesario ejecutarlo para instalar la librería para el manejo de los [EPOS4](#), al final de la instalación se obtiene la confirmación que se presentan en la Figura [D.2](#):

```
sudo bash ./install.sh
```



```
alg_sys@ubuntu:~/EPOS_Linux_Library$ ls
examples include install.sh lib misc
alg_sys@ubuntu:~/EPOS_Linux_Library$ sudo bash ./install.sh
-----
EPOS Command Library 6.3.0.6 installation started
-----
- Remove existing installation [OK]
- Install library into directory: /opt/EposCmdLib 6.3.0.6 [OK]
- Install examples into directory: /opt/EposCmdLib 6.3.0.6 [OK]
- Library system integration [OK]
- Configure device access rights [OK]
udev stop/waiting
udev start/running, process 55502
- Configure user access rights [OK]
-----
EPOS Command Library 6.3.0.6 installed
-----
```

Figura D.2: Instalación de EPOS Library

Cuando la librería se haya instalado con éxito será necesario integrar el archivo “Definition.h” que se encuentra dentro de la carpeta “EPOS\_Linux\_Library” dentro de la carpeta del proyecto que contiene el código que controla los [EPOS4](#) para el manejo de las articulaciones del exoesqueleto.

```
#include "Definitions.h"
```

Con esta librería se puede utilizar la mayoría de los comandos listados dentro del Documento EPOS Command Library [\[52\]](#) dentro del sistema operativo *Linux*. Los comandos que no son compatibles con *Linux* son:

- Comandos de registro de datos (DataRecorder commands)
- Comandos para exportar e importar parámetros (Export/Import parameters commands)
- Comandos relacionados con interfaces gráficas (GUI-related commands)

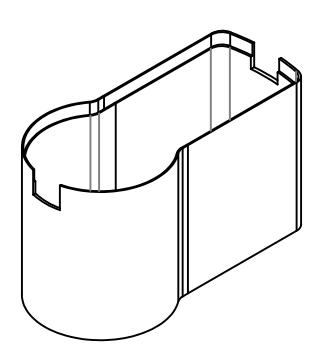
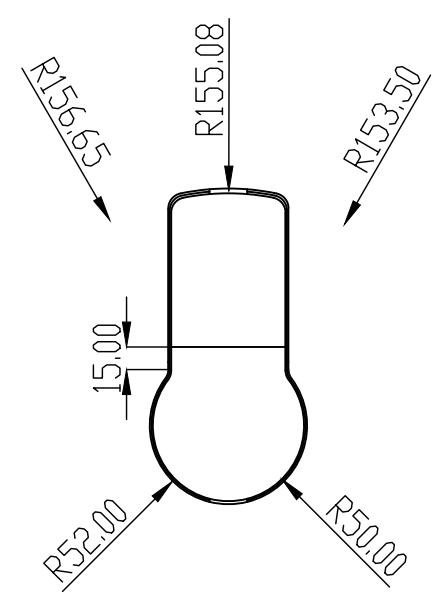
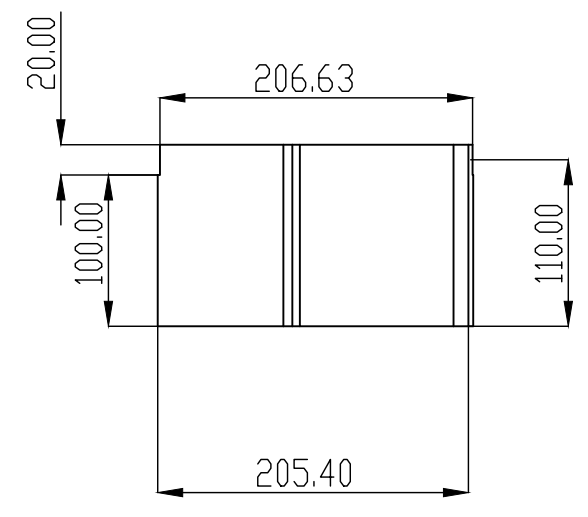
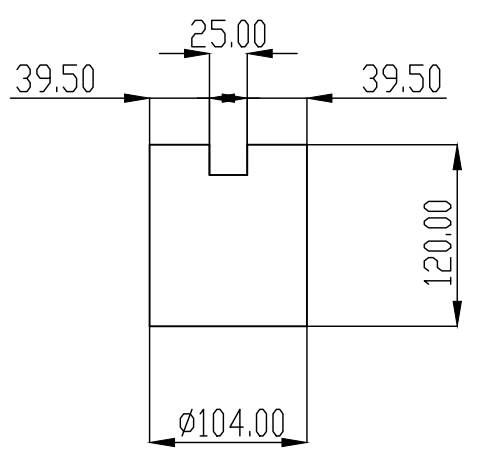




## Apéndice E

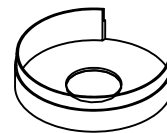
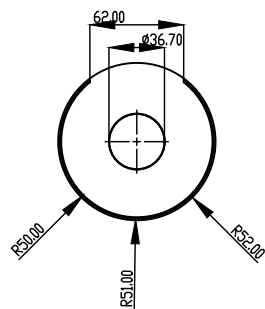
# Planos de los diseños 3D

En esta Sección se muestran los planos en formato CAD para el diseño tanto de las carcasas del exoesqueleto como de los acoples para utilizar el *encoder* externo.

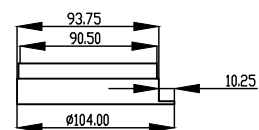
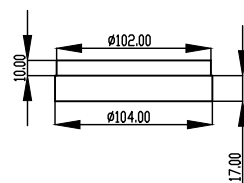
A							A
B							B
C	<div></div>						C
D	<div></div>						D
E							E
F							F
G							G
H		Fecha	Nombre	Firma	UNIVERSIDAD DE CUENCA Ingeniería Electrónica y Telecomunicaciones		H
	Dibujado						
	Revisado						
	ESCALA 1:5	Recubrimiento del motor			PLANO N°:		
					Sustituye a:		
					Sustituido por:		
	1	2	3	4			

A

B



D

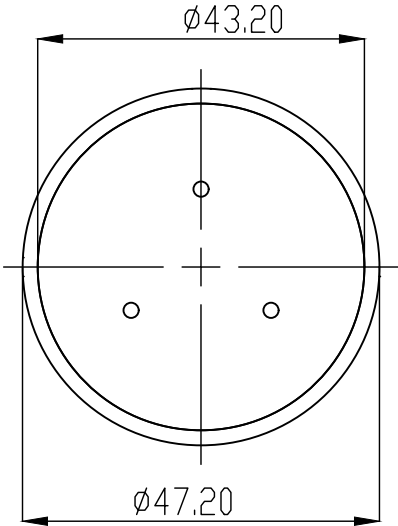
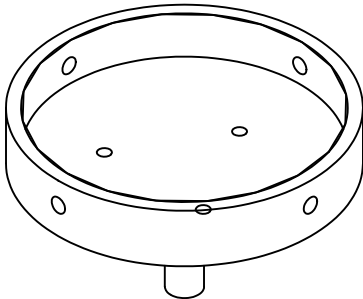
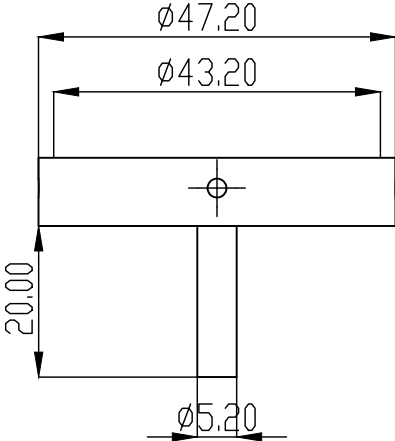
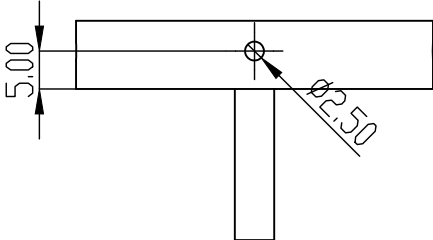


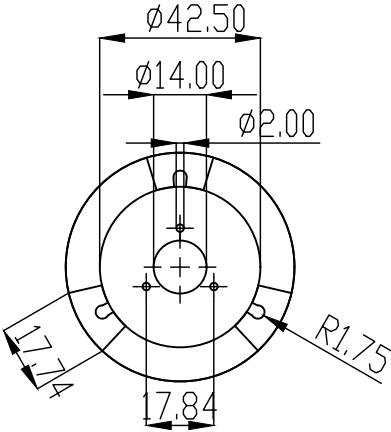
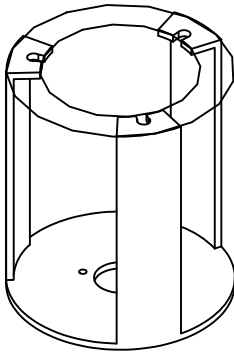
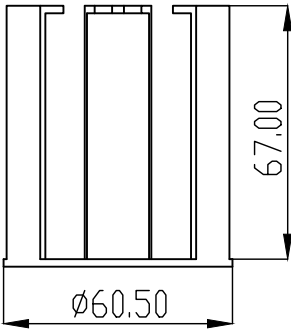
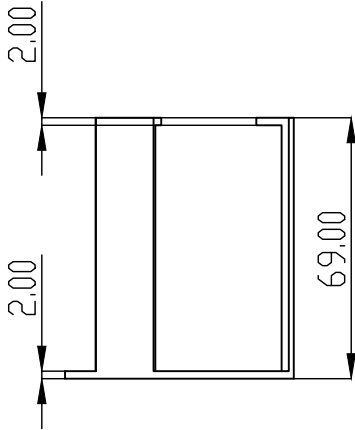
E

F

G

	Fecha	Nombre	Firma	<p style="text-align: center;">UNIVERSIDAD DE CUENCA</p> <p style="text-align: center;">Ingeniería Electrónica y Telecomunicaciones</p>	
Dibujado					
Revisado					
<p style="text-align: center;">ESCALA</p> <p style="text-align: center;">1:5</p>	<p style="text-align: center;">Base del recubrimiento del motor</p>			PLANO N°:	
				Sustituye a:	
				Sustituido por:	
1	2	3	4		

A							A
B							B
C	<div><div></div><div></div></div>						C
D							D
E	<div><div></div><div></div></div>						E
F							F
G							G
H		Fecha	Nombre	Firma	UNIVERSIDAD DE CUENCA Ingeniería Electrónica y Telecomunicaciones		H
	Dibujado						
	Revisado						
	ESCALA 1:5	Acople móvil para encoder				PLANO N°:	
						Sustituye a:	
						Sustituido por:	
	1	2	3	4			

A								A
B								B
C	<div><div></div><div></div></div>							C
D								D
E	<div><div></div><div></div></div>							E
F								F
G								G
H		Fecha	Nombre	Firma	UNIVERSIDAD DE CUENCA Ingeniería Electrónica y Telecomunicaciones			H
	Dibujado							
	Revisado							
	ESCALA 1:5	Acople fijo para encoder				PLANO N°:		
						Sustituye a:		
						Sustituido por:		
	1	2	3	4				

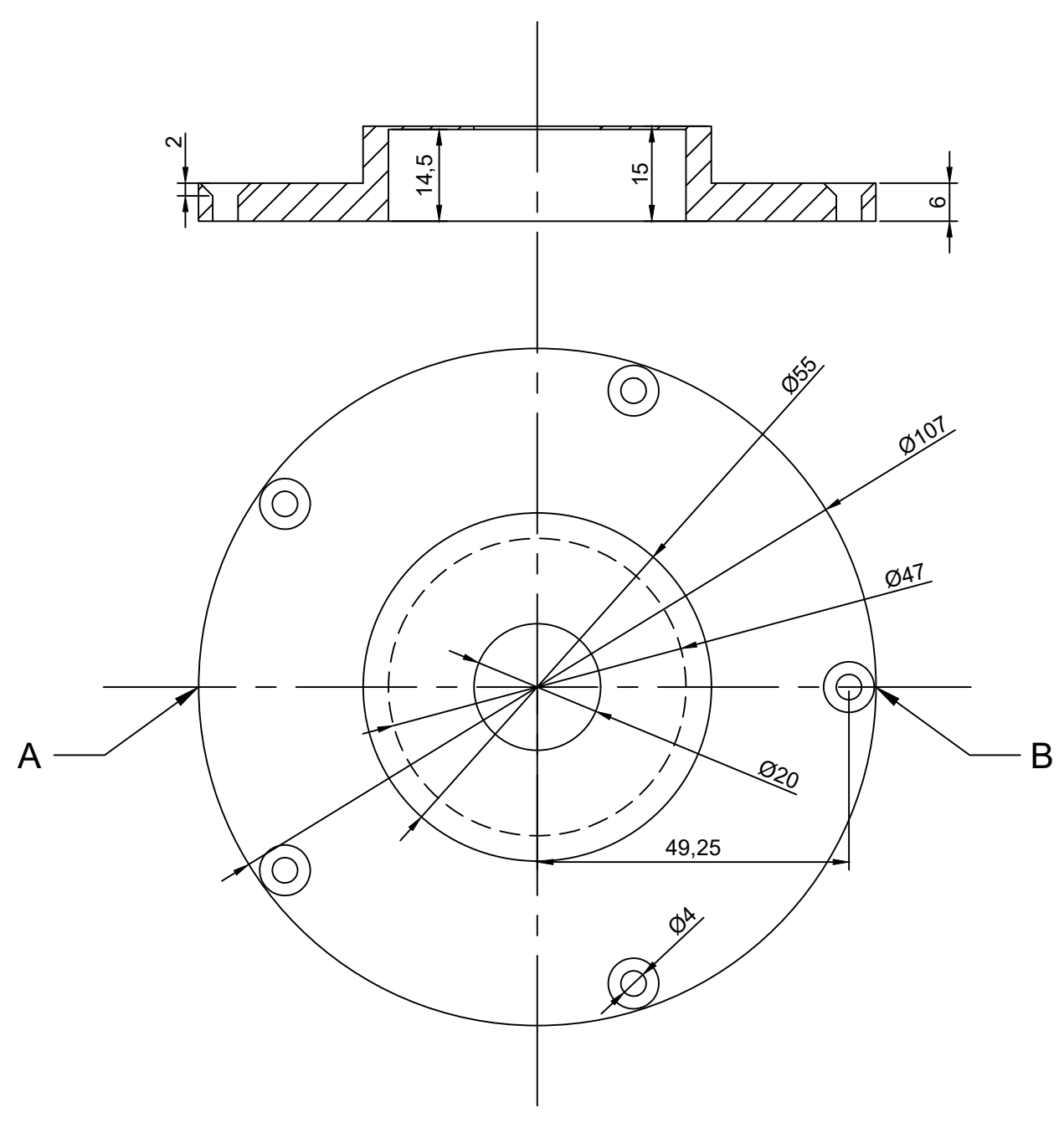




## Apéndice F

# Planos de las cajas reductoras

En esta Sección se muestran los planos en formato CAD para el diseño del sistema reductor.

A						A	
B						B	
C						C	
D						D	
E						E	
F						F	
G						G	
H		Fecha	Nombre	Firma	UNIVERSIDAD DE CUENCA Ingeniería Electrónica y Telecomunicaciones		
	Dibujado						
	Revisado						
	ESCALA 1:1	Soporte superior del reductor cicloidal				PLANO N°:	H
						Sustituye a:	
Sustituido por:							
	1	2	3	4			



A

B

C

D

E

F

G

H

A

B

C

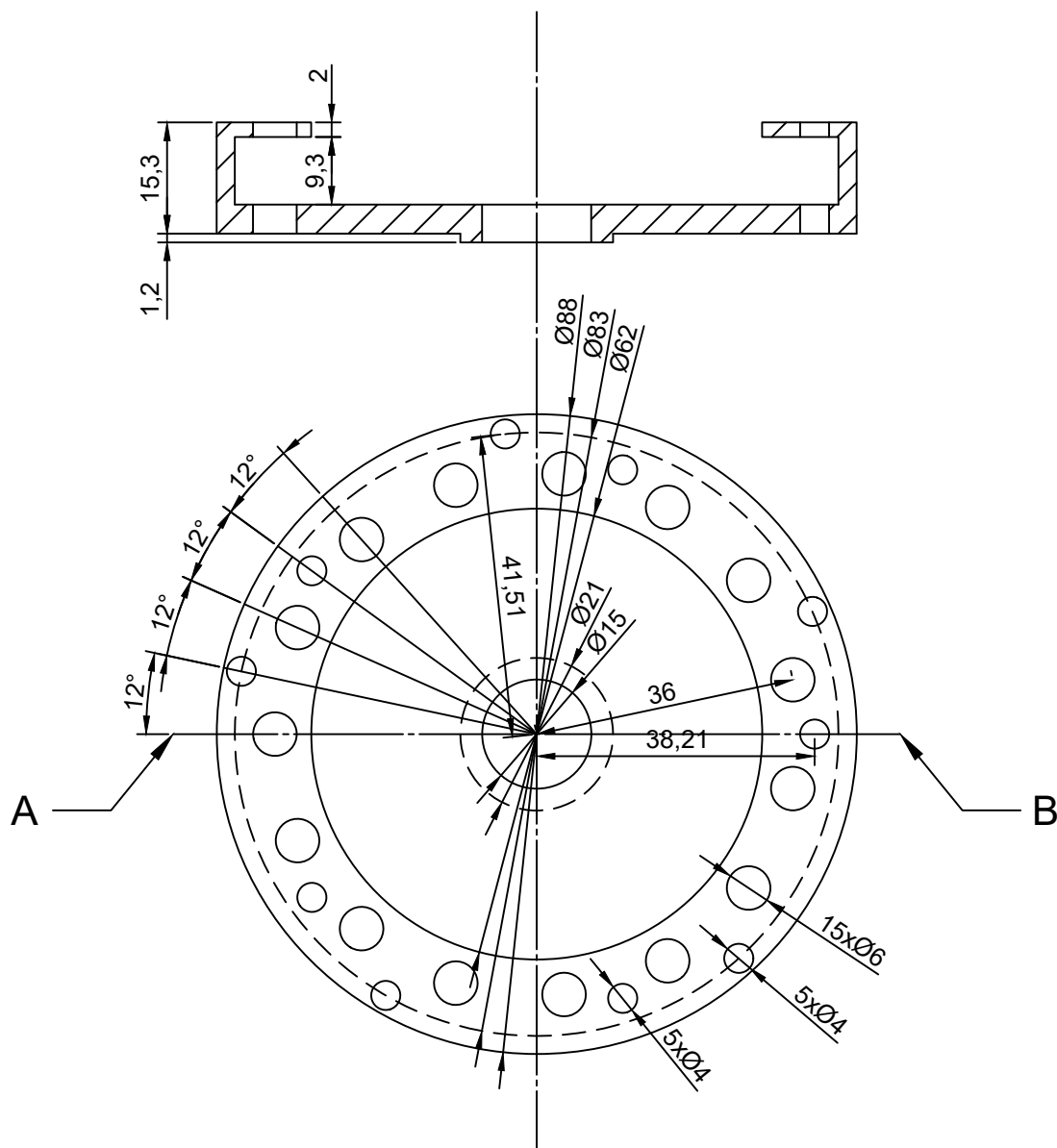
D

E

F

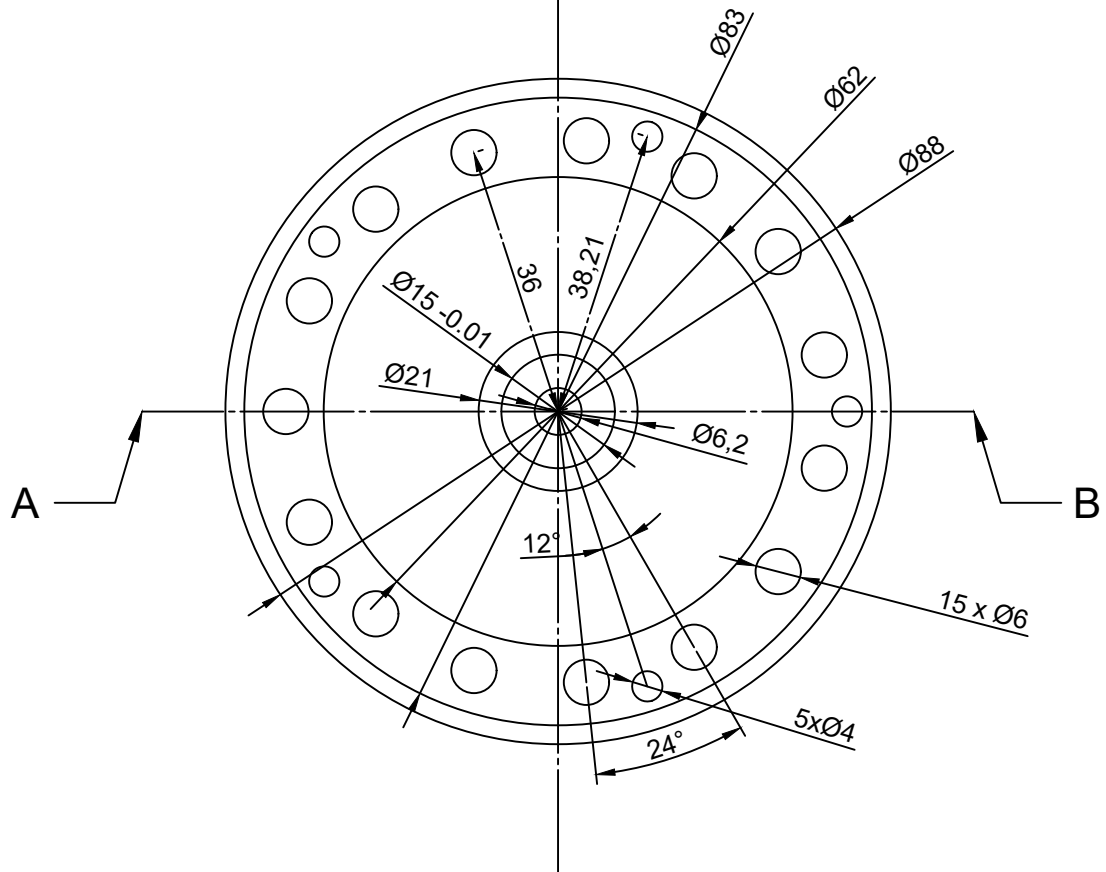
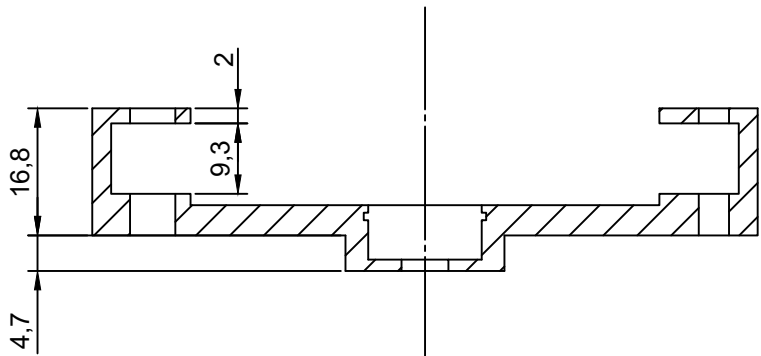
G

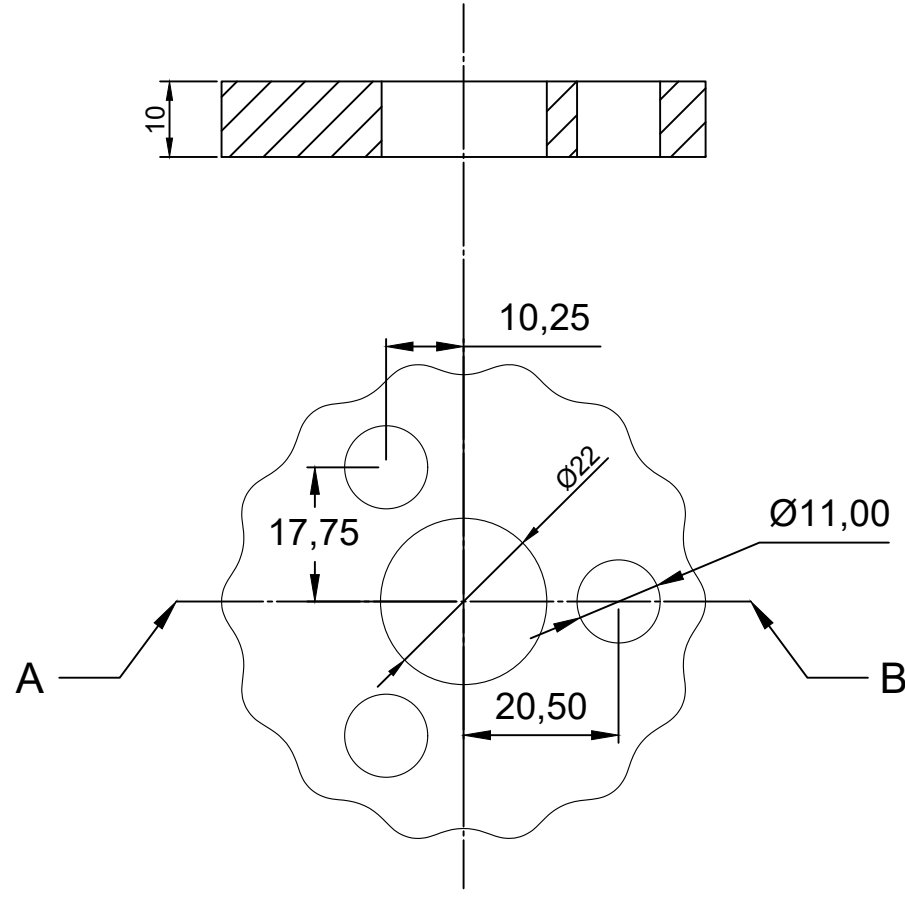
H



	Fecha	Nombre	Firma	<p style="text-align: center;">UNIVERSIDAD DE CUENCA</p> <p style="text-align: center;">Ingeniería Electrónica y Telecomunicaciones</p>	
Dibujado					
Revisado					
<p style="text-align: center;">ESCALA</p> <p style="text-align: center;">1:1</p>	<p style="text-align: center;">Soporte intermedio del reductor cicloidal</p>			PLANO N°:	
				Sustituye a:	
				Sustituido por:	
1	2		3		4

A					A	
B					B	
C					C	
D					D	
E					E	
F					F	
G					G	
H		Fecha	Nombre	Firma	H	
	Dibujado					UNIVERSIDAD DE CUENCA Ingeniería Electrónica y Telecomunicaciones
	Revisado					
	ESCALA 1:1	Soporte inferior del reductor cicloidal				PLANO N°:
					Sustituye a:	
					Sustituido por:	
	1	2	3	4		



A							A
B							B
C							C
D							D
E							E
F							F
G							G
H		Fecha	Nombre	Firma	UNIVERSIDAD DE CUENCA Ingeniería Electrónica y Telecomunicaciones		H
	Dibujado						
	Revisado						
	ESCALA 1:1	Disco cicloidal				PLANO N°:	
						Sustituye a:	
Sustituido por:							
	1	2	3	4			

A							A	
B							B	
C							C	
D							D	
E							E	
F							F	
G							G	
H		Fecha	Nombre	Firma	UNIVERSIDAD DE CUENCA Ingeniería Electrónica y Telecomunicaciones			
	Dibujado							
	Revisado							
	ESCALA 1:1	Eje de salida				PLANO N°:		
						Sustituye a:		
Sustituido por:								
	1	2	3	4				

# Bibliografía

- [1] “GE’s bringing good things, and massive robots, to life - AIVAnet,” <https://www.aivanet.com/2014/01/ges-bringing-good-things-and-massive-robots-to-life/>.
- [2] “Human Universal Load Carrier (HULC),” <https://www.army-technology.com/projects/human-universal-load-carrier-hulc/>.
- [3] “Home use - Rex Bionics,” <https://www.rexbionics.com/rex-for-home-use/>.
- [4] “HAL Lower Limb - Exoskeleton Report,” <https://exoskeletonreport.com/product/hal-lower-limb/>.
- [5] “ReWalk - Exoskeleton Report,” <https://exoskeletonreport.com/product/rewalk/>.
- [6] S. S. d. I. y. N. Científicas, “Un nuevo exoesqueleto se adapta a cada forma de caminar,” Jun. 2017. [En línea]. Disponible: <http://www.agenciasinc.es/Noticias/Un-nuevo-exoesqueleto-se-adapta-a-cada-forma-de-caminar>
- [7] “Los brazos y piernas robóticas con sensores y controlados por la mente son una realidad.” [En línea]. Disponible: <https://www.xataka.com/robotica-e-ia/los-brazos-y-piernas-roboticas-con-sensores-y-controlados-por-la-mente-son-una-realidad>
- [8] A. M. Dollar y H. Herr, “Lower Extremity Exoskeletons and Active Orthoses: Challenges and State-of-the-Art,” *IEEE Transactions on Robotics*, vol. 24, num. 1, pp. 144–158, Feb. 2008.
- [9] L. I. Minchala, F. Astudillo-Salinas, K. Palacio-Baus, y A. Vazquez-Rodas, “Mechatronic Design of a Lower Limb Exoskeleton,” in *Design, Control and Applications of Mechatronic Systems in Engineering*, S. Yildirim, Ed. InTech, May 2017.
- [10] M. Motor, “EPOS4 Application Notes,” p. 112, 2018.
- [11] —, “EPOS4 Communication Guide,” p. 42, 2017.
- [12] —, “EPOS4 Module/Compact 50/8 Hardware Reference,” p. 108, 2016.

- [13] S. V. Thube y T. R. Bobak, "Dynamic analysis of a cycloidal gearbox using finite element method," *AGMA Technical Paper*, pp. 1–13, 2012.
- [14] J. J. Craig, *Robótica*. México: Pearson Prentice Hall, 2006, oCLC: 630216372.
- [15] D. García, "BUS CAN: Estado de buses industriales y aplicaciones." [En línea]. Disponible: <http://www.infopl.net/documentacion/7-comunicaciones-industriales/72-bus-can-estado-de-buses-industriales-y-aplicaciones>
- [16] A. Garrote y R. Bonet, "Lesiones traumáticas de extremidades inferiores," *Offarm*, vol. 22, pp. 52–58, 2003.
- [17] M. A. C. Cardona, F. R. Spitia, y A. B. López, "Exoesqueletos para potenciar las capacidades humanas y apoyar la rehabilitación," *Revista Ingeniería Biomédica*, p. 11, May 2010.
- [18] "REX - Exoskeleton Report," <https://exoskeletonreport.com/product/rex/>.
- [19] "What is a Wearable Robot? - Definition from Techopedia." [En línea]. Disponible: <https://www.techopedia.com/definition/15325/wearable-robot>
- [20] J. L. Pons, Ed., *Wearable robots: biomechatronic exoskeletons*. Hoboken, N.J: Wiley, 2008.
- [21] R. Ceres, J. L. Pons, L. Calderón, y J. Moreno, "La robótica en la discapacidad. Desarrollo de la prótesis diestra de extremidad inferior Manus-Hand," *Revista Iberoamericana de Automática e Informática industrial*, vol. 5, num. 2, pp. 60–68, Abr. 2008. [En línea]. Disponible: <https://polipapers.upv.es/index.php/RIAI/article/view/8276>
- [22] R. López, H. Aguilar, S. Salazar, R. Lozano, y J. A. Torres, "Modelado y Control de un Exoesqueleto para la Rehabilitación de Extremidad Inferior con dos grados de libertad," *Revista Iberoamericana de Automática e Informática industrial*, vol. 11, num. 3, pp. 304–314, 2014. [En línea]. Disponible: <https://polipapers.upv.es/index.php/RIAI/article/view/9443>
- [23] M. P. Jonathan, "Diseño de un exoesqueleto para miembro inferior," Ph.D. dissertation, Instituto politécnico nacional, Mexico D.F., Jun. 2013.
- [24] B. N. T. Cabrera, "DISEÑO Y CONSTRUCCIÓN DE UNA PIERNA EXOESQUELÉTICA PARA LA ASISTENCIA DE LA MARCHA," p. 139.
- [25] B. Chen, H. Ma, L.-Y. Qin, F. Gao, K.-M. Chan, S.-W. Law, L. Qin, y W.-H. Liao, "Recent developments and challenges of lower extremity exoskeletons," *Journal of Orthopaedic Translation*, vol. 5, pp. 26–37, Abr. 2016. [En línea]. Disponible: <https://linkinghub.elsevier.com/retrieve/pii/S2214031X15000716>

- [26] “¿Qué es la biomecánica?” May 2014. [En línea]. Disponible: <http://biomecanicamartinez.com/que-es-la-biomecanica/>
- [27] J. Barreto-Andrade, Adoración Villarroya-Aparicio, Teodoro Contreras-Calle, Vicente Brito-Vásquez, y Esteban Loaiza-Dávila, “Biomecánica de la marcha atlética. Análisis de las presiones plantares durante su desarrollo, revisión actualizada.” [En línea]. Disponible: <http://www.efdeportes.com/efd217/biomecanica-de-la-marcha-atletica.htm>
- [28] F. H. Stengele, “Diseno y construccion de prototipo neumatico de protesis de pierna humana,” May 2008. [En línea]. Disponible: [http://catarina.udlap.mx/u\\_dl\\_a/tales/documentos/lep/herandez\\_s\\_f/](http://catarina.udlap.mx/u_dl_a/tales/documentos/lep/herandez_s_f/)
- [29] A. M. Requena, “Introducción a CAN bus: Descripción, ejemplos y aplicaciones de tiempo real,” p. 98.
- [30] A. Soriano y H. Nazario, “Relación de alumnos que han participado:,” p. 81.
- [31] N. Instruments, “Introducción a CAN - National Instruments.” [En línea]. Disponible: <http://www.ni.com/white-paper/2732/es/>
- [32] A. E. Gutiérrez-Gómez, “Analizador de redes CAN,” Sep. 2017. [En línea]. Disponible: <https://rei.iteso.mx/handle/11117/5151>
- [33] D. Encinas, P. Meilan, J. A. Bava, y R. M. Naiouf, “Protocolo de comunicaciones CAN aplicado a sistemas satelitales y vehículos lanzadores,” p. 9.
- [34] “CAN in Automation (CiA): CANopen.” [En línea]. Disponible: <https://www.can-cia.org/canopen/>
- [35] S. D. Raquel, “CANopen,” Ph.D. dissertation, Universidad de Salamanca, España. [En línea]. Disponible: [http://gro.usal.es/trabajos/Proyecto%20Raquel/web/index\\_archivos/CANopen.pdf](http://gro.usal.es/trabajos/Proyecto%20Raquel/web/index_archivos/CANopen.pdf)
- [36] Siemens, “Interfaz CANopen, Manual de puesta en marcha,” p. 154, Jul. 2016. [En línea]. Disponible: [https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=3&ved=2ahUKEwjBgovpi-bfAhXQslkKHZX2BagQFjACegQIBxAB&url=https%3A%2F%2Fsupport.industry.siemens.com%2Fcs%2Fattachments%2F109740025%2FIH2\\_072016\\_spa\\_es-ES.pdf%3Fdownload%3Dtrue&usg=AOvVaw13mV6wVJbSZ2dcD71g9qOS](https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=3&ved=2ahUKEwjBgovpi-bfAhXQslkKHZX2BagQFjACegQIBxAB&url=https%3A%2F%2Fsupport.industry.siemens.com%2Fcs%2Fattachments%2F109740025%2FIH2_072016_spa_es-ES.pdf%3Fdownload%3Dtrue&usg=AOvVaw13mV6wVJbSZ2dcD71g9qOS)
- [37] “Control Epos Page.” [En línea]. Disponible: <https://www.maxonmotor.es/maxon/view/content/epos-detailsite>
- [38] J. A. Medrano, I. Zendejas-Hernandez, y R. Sandoval-Rodríguez, “Diseño de un Exoesqueleto de Miembros Inferiores para la Movilidad de Personas con Paraplejía,” in *Congreso Mexicano de Robótica*, México, 2014, p. 8.



- [39] R. J. Schilling, *Fundamentals of Robotics: Analysis and Control*. Englewood Cliffs, N.J: Prentice Hall, 1990.
- [40] T. Yoshikawa, *Foundations of Robotics: Analysis and Control*. Cambridge, Mass: MIT Press, 1990.
- [41] S. R. Schmid, B. J. Hamrock, y B. O. Jacobson, *Fundamentals of Machine Elements*, 3ra ed. Boca Raton, Fla.: CRC Press, 2014, oCLC: 897455902.
- [42] B. Borislavov, I. Borisov, y V. Panchev, *Design of a Planetary-Cyclo-Drive Speed Reducer : Cycloid Stage, Geometry, Element Analyses*. Linnaeus University, School of Engineering, 2012.
- [43] Dassault Systèmes, “3d CAD Design Software.” [En línea]. Disponible: <https://www.solidworks.com/>
- [44] D. Dudley, *Manual de Engranajes: Diseño, Manufactura y Aplicación de Engranajes*. Compañía Editorial Continental, 1973.
- [45] Robo 3D, “Robo R2 3d Printer | ROBO 3d.” [En línea]. Disponible: <https://robo3d.com/products/robo-r2>
- [46] P.-A. Willems, B. Schepens, y C. Detrembleur, “Marcha normal,” *EMC - Kinesiterapia - Medicina Física*, vol. 33, num. 2, pp. 1–29, Abr. 2012.
- [47] “PiCAN 2 - CAN Interface for Raspberry Pi 2/3.” [En línea]. Disponible: <https://copperhilltech.com/pican-2-can-interface-for-raspberry-pi-2-3/>
- [48] M. Motor, “Motor Maxon EC45-70w 397172.” [En línea]. Disponible: [https://www.maxonmotor.com/medias/sys\\_master/root/8825435127838/17-EN-267.pdf](https://www.maxonmotor.com/medias/sys_master/root/8825435127838/17-EN-267.pdf)
- [49] —, “Motor Maxon EC90 323772.” [En línea]. Disponible: [https://www.maxonmotor.com/medias/sys\\_master/root/8825435389982/17-EN-271.pdf](https://www.maxonmotor.com/medias/sys_master/root/8825435389982/17-EN-271.pdf)
- [50] —, “Encoder MILE 462004.” [En línea]. Disponible: [http://www.ensatek.com.tr/image/urun/encoder/1\\_MILE\\_256.pdf](http://www.ensatek.com.tr/image/urun/encoder/1_MILE_256.pdf)
- [51] —, “Encoder MILE 453233.” [En línea]. Disponible: [http://www.ensatek.com.tr/image/urun/encoder/3\\_MILE\\_512\\_B.pdf](http://www.ensatek.com.tr/image/urun/encoder/3_MILE_512_B.pdf)
- [52] —, “EPOS-Command-Library-En.” [En línea]. Disponible: [https://www.maxonmotor.com/medias/sys\\_master/root/8823917281310/EPOS-Command-Library-En.pdf](https://www.maxonmotor.com/medias/sys_master/root/8823917281310/EPOS-Command-Library-En.pdf)